Adaptive neuro-fuzzy inference system for flood forecasting in a large river system

Nguyen, Khac Tien Phuoc

2012

Nguyen, K.-T. P. (2012). Adaptive neuro-fuzzy inference system for flood forecasting in a large river system. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/50919>

<https://doi.org/10.32657/10356/50919>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM
FOR FLOOD FORECASTING IN A LARGE RIVER
SYSTEM**

**NGUYEN KHAC TIEN PHUOC
SCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING
2012**

**ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM
FOR FLOOD FORECASTING IN A LARGE RIVER
SYSTEM**

NGUYEN KHAC TIEN PHUOC

School of Civil and Environmental Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirement for the degree of
Doctor of Philosophy.

2012

ACKNOWLEDGEMENTS

I would like to express my special thanks to my supervisor, Assistant Professor Dr. Chua Hock Chye Lloyd. He provided supervision, asked questions and frequently discussed with me my research topic. He provided me guidance and directed my research resulting in the material contained in this thesis. Without his support and encouragement, I would not have been able to complete this work.

I would also like to thank the Regional Flood Management and Mitigation Centre – Mekong River Commission (RFMMC - MRC) for providing the measured data which was essential material for my research. I especially wish to express my gratitude to Dr. Lam Hung Son (Programme Coordinator for Flood Management and Mitigation Programme – Mekong River Commission), Mr. Nicolaas Bakker (Chief Technical Advisor), Mr. Nguyen Tien Kien (Operational Meteorologist), Dr. Katry Phung (Hydrologist), Dr. Pichaid Varoonchotikul (River Flood Forecasting Expert) and other staff members for sharing with us their knowledge and experience with the operational flood forecasting system at RFMMC - MRC and for the valuable discussions on our results in our frequent visits to Phnom Penh.

I must also acknowledge Nanyang Technological University for generously providing the PhD scholarship for the 4 years of study at NTU. In addition, I also thank the DHI-NTU Water & Environment Research Centre and Education Hub for providing funding support for my field visits. Acknowledgements also go to my friends and colleagues who have assisted and encouraged me during my 4 years in NTU.

Finally, I would like to express my heartfelt gratitude to my family members, my wife and my sons for their love, understanding and constant encouragement. That was a great support for me which enabled me to complete my study.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
TABLE OF CONTENTS	ii
ABSTRACT... ..	v
LIST OF FIGURES	vii
LIST OF TABLES	xi
LIST OF ABBREVIATIONS.....	xii
LIST OF SYMBOLS	xiii
CHAPTER 1: INTRODUCTION.....	1
1.1. Flood Forecasting for Large River Systems.....	1
1.2. Real-Time Flood Forecasting.....	2
1.3. Aim and Objectives of the Study.....	7
1.4. Structure of the Report.....	8
CHAPTER 2: LITERATURE REVIEW.....	10
2.1. Introduction	10
2.2. Physically-Based Modelling Approach	12
2.2.1. The Unified Run-off Basin Simulation (URBS) Model.....	16
2.3. Statistical Method with Multiple Linear Regression Model.....	19
2.4. Data-driven Modelling Approaches.....	21
2.5. Data Assimilation.....	26
2.6. Data Driven Models - Challenges Faced for Large Rivers.....	28
CHAPTER 3: METHODOLOGY.....	29
3.1. Introduction.....	29
3.2. Adaptive Network with Structure and Basic Learning Rule.....	30
3.3. Hybrid Learning with Batch (Off-Line) Mode.....	33
3.4. Hybrid Learning with Pattern (On-Line) Mode.....	36
3.5. Fuzzy logic, Fuzzy rules and Fuzzy Inference Systems.....	37
3.5.1 Fuzzy Logic.....	37

3.5.2 Fuzzy IF-THEN Rules.....	39
3.5.3 Fuzzy Inference Systems.....	40
3.6. ANFIS: Adaptive Neuron-Fuzzy Inference System.....	43
3.7. Real-Time Updating Techniques.....	46
3.7.1. Recursive Autoregression Models.....	47
 CHAPTER 4: DATA USED.....	 50
4.1. Floods in the Lower Mekong River	50
4.2. Review of Flood Forecasting Models for the Lower Mekong	53
4.3. Hydro-Meteorological Data.....	54
4.3.1. Gauging Stations under Study.....	55
4.4. Evaluation of Forecast Accuracy.....	56
4.5. Data Normalization	59
 CHAPTER 5: THE DATA-DRIVEN APPROACH AS A FLOOD FORECASTING MODEL.....	 61
5.1. Introduction.....	61
5.2. ANFIS Model Development.....	63
5.2.1. Data Division.....	63
5.2.2. Input–Output Selection.....	64
5.2.3. Model Structure Selection and Calibration.....	67
5.3. Other Models used for Comparison with ANFIS Models.....	69
5.3.1. Multiple Linear Regression Models.....	69
5.3.2. Multi-Layer Perceptron Neural Network.....	70
5.3.3. URBS Mode.....	71
5.4. Output Error Correction Using Autoregressive Models.....	71
5.5. Results and Discussions.....	75
5.5.1. A Comparison between ANFIS and MLR, MLP Models.....	75
5.5.2. Comparison between ANFIS and URBS Models.....	78
5.5.3. ANFIS Results with Updating.....	81
5.6. Correlation Analyses.....	86
5.7. Rising Limb.....	89

5.8. Conclusions.....	93
CHAPTER 6: WATER LEVEL FORECASTING IN LARGE RIVER SYSTEMS: INPUT SELECTION FOR DATA-DRIVEN MODELS	95
6.1. Introduction.....	95
6.2. Input Selection of ANFIS Models.....	98
6.3. Model Structure Selection and Implementation.....	101
6.4. Results and Discussion.....	104
6.4.1. Forecast Results for Model M2.....	104
6.4.2. Forecast Results for Model M3.....	105
6.4.3. Forecast Results for Model M4.....	107
6.5. Conclusions.....	109
CHAPTER 7: APPLICATION OF ON-LINE MODELS FOR FLOOD FORECASTING.....	112
7.1. Introduction.....	112
7.2. On-line ANFIS Model.....	114
7.3. Water Level Forecasting Using ANFIS with On-line Training.....	115
7.4. Results and Discussion.....	117
7.4.1. Scenario 1.....	117
7.4.2. Scenario 2.....	122
7.4.3. Analysis of the Rising and Falling Phases of Flood Hydrographs.....	123
7.5. Conclusions.....	139
CHAPTER 8: CONCLUSIONS.....	140
PUBLICATIONS.....	144
REFERENCES.....	145

ABSTRACT

Large river systems provide essential water resources and economic benefits for a significant number of people. However, large floods that occur less frequently (e.g., 25 to 100-year floods) can cause significant loss of lives and property. Therefore, flow forecasting, which aims to provide accurate forecasts of flow and/or level of a river is an essential tool for flood plain management and flood mitigation purposes. New approaches in computational intelligence tools are now being applied to build effective forecasting models. This research investigated the application of the data driven modeling approach for flood forecasting in the Lower Mekong, which was chosen as a case study. The Adaptive Neuron-Fuzzy Inference System or ANFIS was chosen for this purpose. ANFIS is a modern approach in Computational Intelligence which combines the learning and reasoning capabilities of the Artificial Neural Network and Fuzzy Inference System techniques, respectively.

In order to develop a flood forecasting model based on ANFIS, it was important to clarify the link between hydraulic/hydrologic factors (such as flood travel times and the catchment response to rainfall) and ANFIS model performance, as this would provide important information for model input selection. River flows at various stations along a large river can vary in terms of the flow contributions from upstream stations, rainfall and lateral inflows from sub-basins immediately upstream of the station of interest. Therefore, the present study provided an analysis on how these factors influence ANFIS model performance by developing a series of ANFIS models for three stations along the Lower Mekong where the flows at the stations selected differed in terms of the contributions by flow from upstream stations along the main river, laterals and rainfall from the local sub-basin. In such a scenario, the choice of input variables can significantly affect ANFIS model performance. The results obtained showed that the inclusion of water levels from the station of interest and the immediate upstream station were sufficient for 1- and 3-lead day water level forecast. The addition of rainfall information beside water levels was not necessary. The ANFIS models developed in the study performed well for 1- and 3-lead-day water level forecasts when compared to the benchmark adopted by Mekong River Commission (MRC). In addition, ANFIS model results

were superior to the results provided by the Unified Run-off Basin Simulation (URBS) conceptual runoff routing model, which is currently implemented by the Regional Flood Management and Mitigation Centre of Mekong River Commission for operational forecasts. For longer forecast lead times, the inclusion of rainfall inputs was necessary. This was especially true in the case of Thakhek station where the contribution from rainfall in the local sub-basin was significant. The improvements at other stations, where flows contributed from rainfall in the local sub-basin were less significant, were modest. This study also highlights the difficulties faced by both the data driven and conceptual models for 5-lead-day forecasts.

Two approaches to improve ANFIS models results were investigated. The first approach was to employ an updating procedure using the partially recursive output error prediction method. Considerable improvements to 3-, 4- and 5-lead-day forecasts were obtained when ANFIS model results were corrected by the updating procedure. The updating procedure uses the Recursive Auto-regression model as a time-series prediction model and uses the 2-lead-day ANFIS model results to initiate the error estimation process. The second approach was to adopt an on-line learning algorithm in training the ANFIS model. With this approach, the ANFIS models' parameters are updated continuously whenever new data is available. The ANFIS model trained using a local learning algorithm was found to significantly improve ANFIS model results; the on-line ANFIS models performed better than off-line ANFIS models for all forecast lead-days and at all the stations under study. In addition, the model trained with the local learning algorithm required less training data, compared to the off-line model. This is useful especially where data is lacking. In addition, even when the off-line model was trained using a sufficiently long training data set, the on-line was found to provide either comparable or better results. However, the results showed that the on-line training algorithm may not improve the accuracy of forecasts during the rising phase of flood events, although greater attenuation of forecast errors can be achieved in the falling phase.

LIST OF FIGURES

		Page
Figure 2.1	The typical frame work for stream flow simulation with two parts: (1) Rainfall-Runoff and (2) Flow routing	14
Figure 3.1	An adaptive network with two inputs and two outputs	30
Figure 3.2	Examples of membership functions	38
Figure 3.3	A diagram of a Fuzzy Inference System (Jang, 1993)	40
Figure 3.4	An example of Tagaki-Sugeno-Kang fuzzy inference	42
Figure 3.5	A schematic of ANFIS with 2 rules for a system consisting of 2 inputs and 1 output according to Jang (1993)	43
Figure 4.1	(a) Location map for the Mekong Basin, (b) Rainfall distribution for the Lower Mekong Basin, (c) Sub-basin 1: 16, 906 (km ²), (d) Sub-basin 2: 10,407 (km ²), (e) Sub-basin 3: 119,707 (km ²), (f) Sub-basin 4: 28,815 (km ²). Gauging stations on main stream, tributaries, rainfall stations are shown in (c) – (f). Source MRC (2004), (2006)	51
Figure 4.2	The increase of the Mean Annual Flow along the Lower Mekong from Chiang Sean to the delta in term of volume and percentage	56
Figure 5.1	Correlogram of the water level time series at Pakse: (a) Autocorrelation and (b) Partial autocorrelation	65
Figure 5.2	The relationship between the water levels at Pakse and upstream stations: (a) cross-correlation and (b) RMSE based on shifted time series	67
Figure 5.3	Membership functions identified for water level at Pakse in ANFIS model	69
Figure 5.4	Training and validation errors corresponding to number of hidden nodes in the MLP neural network	71
Figure 5.5	The output error prediction procedures for the ANFIS ₅ model: (a) AR _a and (b) AR _b procedures	74
Figure 5.6	Model performance comparison between linear regression, ANFIS and ANN models for 5-lead-day water level forecasting at Pakse.	76
Figure 5.7	A comparison between observed water level at Pakse and 5-lead-day forecast given by ANN and ANFIS models.	77

Figure 5.8	MAE calculated on the one- to five-lead-day forecasts for entire wet season (with and without updating) for (a) the training data set and (b) the testing data set	78
Figure 5.9	MAE calculated on the one- to five-lead-day forecasts for high water level (with and without updating) for (a) the training data set and (b) the testing data set	79
Figure 5.10	A comparison of performance between Naïve, URBS and ANFIS: (a) COE, (b) MPAE, (c) MAE and (d) RMSE	80
Figure 5.11	A comparison of AR_a , RAR_a , AR_b and RAR_b updating procedures	82
Figure 5.12	A comparison between actual and estimated values obtained for the error $e_{t-3}(t + 5)$ by (a) RAR_a and (b) RAR_b	83
Figure 5.13	A comparison of the ANFIS model results without and with updating procedure for 1- to 5-lead-day forecasts for (a) the training data set and (b) the testing data set	84
Figure 5.14	Measured and computed water level time series of ANFIS ₅ and ANFIS ₅ + RAR_b for five-lead-day forecasts in (a) year 2000 and (b) year 2001	85
Figure 5.15	Plots of relationship between (a, b) the errors $e_{t-1}(t + 1)$ and $e_t(t + 1)$ for ANFIS ₁ , (c, d) the errors $e_{t-1}(t + 3)$ and $e_t(t + 3)$ for ANFIS ₃ and (e, f) $e_{t-1}(t + 5)$ and $e_t(t + 5)$ for ANFIS ₅	87
Figure 5.16	Autocorrelation and partial autocorrelation of ANFIS models' training errors at (a) one-lead-day, (b) two-lead-day, (c) three-lead-day, (d) four-lead-day and (e) five-lead-day forecasts	88
Figure 5.17	(a) MAE (m), and (b) MPAE (%) of ANFIS ₁ , ANFIS ₃ and ANFIS ₅ for one-, three- and five-lead-day water level forecasts of the rising limbs for training (events 1, 2 and 3), validation (events 4 and 5) and testing (events 6, 7 and 8) events	90
Figure 5.18	ANFIS model forecast accuracy as a function of ΔH	92
Figure 5.19	Comparison between the one-, three- and five-lead-day water level forecasts with observed water levels of the rising limbs for validation (events 4 and 5) and (events 6, 7 and 8) testing events	92
Figure 6.1	Correlogram of the water level time series at Thakhek, Pakse, and Kratie: (a) autocorrelation (ACF), and (b) partial autocorrelation (PACF) of water level time series	98
Figure 6.2	Cross correlation between water levels at (a) Thakhek (b) Pakse (c) Kratie and water levels at upstream stations	99

Figure 6.3	Cross correlation between average areal rainfall in sub-basin 1 and water level at Thakhek; sub-basin 2 and water level at Pakse; sub-basin 3 and water level at Pakse; sub-basin 4 and water level at Kratie. (a) 3 point moving average, (b) 5 point moving average and (c) 7 point moving average.	101
Figure 6.4	Model performance for Model M2. (a) Thakhek, (b) Pakse, and (c) Kratie.	105
Figure 6.5	Comparison between M3 and M2 models. (a) Thakhek, (b) Pakse, and (c) Kratie based on RMSE, CE and PI.	106
Figure 6.6	Comparison between M4 and M2 models. (a) Thakhek, (b) Pakse, and (c) Kratie.	109
Figure 7.1	The hybrid learning paradigm for on-line learning applications	114
Figure 7.2	Comparison between on-line and off-line ANFIS models for 1-lead-day water level forecasts at (a) Thakhek, (b) Pakse, and (c) Kratie	119
Figure 7.3	Comparison between on-line and off-line ANFIS models for 3-lead-day water level forecasts at (a) Thakhek, (b) Pakse, and (c) Kratie	120
Figure 7.4	Comparison between on-line and off-line ANFIS models for 5-lead-day water level forecasts at (a) Thakhek, (b) Pakse, and (c) Kratie	121
Figure 7.5	Mean Absolute Error of M4b ₁ and M4b ₂ models for training, validation and test datasets at (a) Thakhek, (b) Pakse, and (c) Kratie	122
Figure 7.6	Definition of the rising and falling limbs of flood events recorded at Thakhek: (a) Year 2000 from 01 Jun 2000 to 31 Oct 2000, and (b) Year 2001 from 01 Jun 2001 to 41 Oct 2001	125
Figure 7.7	Mean Absolute Error of M4b ₁ and M4b ₂ for the rising limbs of 8 flood events at (a) Thakhek, (b) Pakse, and (c) Kratie	127
Figure 7.8	Evaluation of off-line and on-line models for the rising limbs of flood events in the validation and test data set at: (a) Thakhek, (b) Pakse, and (c) Kratie	129
Figure 7.9	Comparisons between observed and simulated water levels for (a.) 1-lead-day, (b.) 3-lead-days, and (c.) 5-lead-days forecasts	132
Figure 7.10	Mean Absolute Error of M4b ₁ and M4b ₂ for the falling limbs of 8 flood events at (a) Thakhek, (b) Pakse, and (c) Kratie	134

Figure 7.11	Evaluation of off-line and on-line models for falling limbs of validation and test flood events at: (a) Thakhek, (b) Pakse, and (c) Kratie	135
Figure 7.12	Comparison of water level time series at Thakhek, Pakse and Kratie for (a.) 1-lead-day, (b.) 3-lead-days, and (c.) 5-lead-days forecasting	138

LIST OF TABLES

	Page
Table 1.1 The properties of various types of real-time flood forecasting system (Feldman, 1994)	4
Table 4.1 Summary of MRC Hydro-meteorological stations	54
Table 4.2 Near real-time hydro-meteorological gauging stations managed by the MRC	55
Table 4.3 Water level regimes, alarm and flood levels for Thakhek, Pakse, and Kratie	59
Table 5.1 Statistical properties of water levels used in the training, validation, and testing datasets. (N = number of samples, Min, Max = minimum and maximum values, respectively, S.D. = standard deviation, Cv = coefficient of variation and Cs = skewness coefficient)	64
Table 5.2 Comparison between ANN and ANFIS during three flood events in validation and test data set.	77
Table 5.3 MAE (m) of ANFIS ₅ output results with/without updating by the RAR _b procedure for the 2000 and 2001 floods	86
Table 5.4 Data on rising limbs for flood events during the period 1993 to 2003	90
Table 6.1 Inputs used and structure of ANFIS models adopted	103
Table 7.1 Structure of M4b ₁ and M4b ₂ models	116
Table 7.2 Data for (a.) Rising and (b.) Falling limbs for flood events at Thakhek during the period 1993 to 2003	124
Table 7.3 The range (min, max) of Δ MAE (m) between M4b ₁ and M4b ₂ models for the rising limb of the validation and test events	126
Table 7.4 The range (min, max) of Δ MAE (m) between M4b ₁ and M4b ₂ models for the falling limb of the validation and test events	133

LIST OF ABBREVIATIONS

ANFIS	Adaptive Neuro-Fuzzy Inference System
ANNs	Artificial Neural Networks
AR	Auto-Regression Model
ARMA	Auto-Regression with Moving Average Model
ARX	Auto-Regression Model with Exogenous Inputs Model
DEM	Digital Elevation Model
DHI	Danish Hydraulic Institute
FIS	Fuzzy Inference System
GIS	Geographic Information System
HEC-RAS	Hydrologic Engineering Center - River Analysis System
MF	Membership Function
MLP	Multilayer Perceptron Neural Network
MRC	Mekong River Commission
NARX	Non-Linear Auto-Regression Model
NFIS	Neuro-Fuzzy Inference Systems
NOAA	National Oceanic and Atmospheric Administration
RAR	Recursive Auto-Regression Model
RBF	Radial Basis Function Network
RFMMC	Regional Flood Management and Mitigation Center
SRE	Satellite Rainfall Estimates
SSARR	Streamflow Synthesis and Reservoir Regulation Model
URBS	Unified Run-off Basin Simulation
WMO	World Meteorological Organization

LIST OF SYMBOLS

A	Area of a sub-catchment (m^2)
CE	Coefficient of Efficiency
E	Total error measure
E_p	Error measure for the p^{th} sample in training data
F	Fraction of forested sub-catchment
H	Water level (m)
H_{Pakse}	Water level at Pakse gauging station (m)
H_{Thakhek}	Water level at Thakhek gauging station (m)
H_{Kratie}	Water level at Kratie gauging station (m)
k_c	Non-linear routing constant
LSE	Least Squares Estimate
MAE	Mean Absolute Error (m)
n	Manning channel roughness ($\text{s/m}^{1/3}$)
$O_{m,p}$	The m^{th} component of actual output vector produced by the presentation of the p^{th} input vector.
P_{w3}	Areal average rainfall with moving average by a window of 3 days (mm)
P_{w5}	Areal average rainfall with moving average by a window of 5 days (mm)
P_{w7}	Areal average rainfall with moving average by a window of 7 days (mm)
PI	Persistence Index
Q	Discharge (m^3/s)
RMSE	Root Mean Square Error (m)

S	Catchment and channel storage (m^3)
S_c	Channel slope
T	Travel time of flow (hour)
$T_{m,p}$	The m^{th} component of p^{th} target output vector
U	Fraction urbanisation of sub-catchment
α	The generic parameter of an adaptive network
$\Delta\alpha$	The difference in α
$\partial E_p / \partial O$	Derivative of training error E_p of the p^{th} sample respect to the output O
$\partial E_p / \partial \alpha$	Derivative of training error E_p of the p^{th} sample respect to the parameter α
$\partial E / \partial \alpha$	Derivative of total error E respect to the parameter α

CHAPTER 1

INTRODUCTION

1.1. Flood Forecasting for Large River Systems

Floods in rivers are natural phenomena that occur when water levels in river channels lead to overspill of natural banks or artificial embankments and subsequently inundate the surrounding flood plains (White, 2000). River floods usually occur after heavy rainfalls over river basins during the wet season. This causes increased runoff into upstream reaches of rivers, and when downstream water levels are increased to very high levels, the rivers may overspill the natural banks or artificial embankments or even break dike systems at some locations, resulting in flows onto floodplains.

Large rivers refer to rivers having drainage areas, annual average water discharge and the length of main stream of the order 10^6 km², 10^9 m³ and 10^3 km, respectively. Some examples include the Amazon River in South America, Changjiang River in China, Mississippi River in North America, and the Mekong River in Asia, with lengths of 6,000 km, 6,300 km, 6,000 km, and 4,800 km, drainage basins of 5.9×10^6 km², 1.9×10^6 km², 3.22×10^6 km² and 0.79×10^6 km², and annual average water discharges of $6,300 \times 10^9$ m³, 900×10^9 m³, 530×10^9 m³, and 470×10^9 m³, respectively (Gupta, 2007). Large river systems provide essential water resources for a significant number of people and opportunities for development on the floodplains. This, on one hand, provides much social and economic benefits, but on the other hand, increases exposure to flood risk. Floods may occur quite often in the floodplains of a large river. However, different magnitude of floods may cause different impacts. Hence, annual river floods with regular magnitudes can be beneficial since they provide water and nutrient rich sediments that are useful for agriculture; however, very large floods occur less frequently (e.g., 25 to 100-year floods) but can cause significant loss of lives and property. Consequently, the reconstruction costs are much higher and require a longer recovery time.

Accurate forecasting of water levels and/or discharge is a key requirement for the successful implementation of flood management and mitigation in a large river system. For example, accurate water level forecasts can help reservoir operators to manage reservoirs more efficiently and hence reduce flood risks. Water level forecasts also provide early warning of impending floods with the potential of saving human lives and property. Therefore, in large river basins, a flood forecasting system is an essential tool for flood management and mitigation. In such a system, hydrological data are collected either in daily or sub-daily intervals and transmitted to a forecast centre in real-time. At the forecast centre, data will be pre-processed then fed into a flood forecasting model (which can be a combination of a number of sub-models) to provide reliable forecasts of flow and stage of a river at key locations along a river.

1.2. Real-Time Flood Forecasting

A long-term river flow prediction model aims to provide long-term information for the purpose of planning or design. Real-time river flow forecasting aims to provide accurate forecasts of flow and/or level of a river in the short-term (i.e., hours to days of time steps ahead) usually during in the wet season which is essential information for flood plain management and flood mitigation purposes. A flood forecasting system usually consists of a system for the collection and transmission of hydrological data to a forecast centre where data will be pre-processed and then input to a flood forecasting model to provide reliable forecasts of flow and stage of a river at key locations along the river during a flood. In addition, a flood forecasting model usually uses feedback information (i.e., the deviation of forecast values from observed ones) to improve on its forecasting performance as soon as the measurements of the current river flow are received (WMO, 1992). Flood forecasting is an essential component of real-time operation of hydrosystems, flood warning, and decision making in order to mitigate flood damages.

Advances in information and communication technologies have contributed to important roles in the improvement of data collection, transmission and data processing, and modeling in real-time flood forecasting systems. New approaches in

hydroinformatics are being applied to build effective hydrological forecasting models, which are the core of these systems. The model should encapsulate sufficient characteristics of the river system (i.e., rainfall-runoff and flood routing characteristics). It also needs to be able to simulate past flood events well. Moreover the model should be able to present well the response of the river basin to each storm event. To develop an effective real-time flood forecasting system, various issues should be considered. The important issues can be identified as follows:

(i) Data collection, data transmission and data preparation

Input data is essential for a forecast system. The more data available, the greater the level of accuracy of forecast. Therefore, the first issue of a real-time flood forecasting system is to collect sufficient data and to transmit the data to data stores where the data is ready for use by the model. The hydrological data (e.g., rainfall, discharge, water level, etc) collected at gauges should have sufficient resolution to represent the variation in time and space of rainfalls and the variation of river flow and water level from upstream to downstream. The data must also be transmitted to the forecasting center as soon as possible in real-time since the usefulness of forecasts is measured in terms of its accuracy and forecast lead-time. Nowadays, advances in remote sensing techniques such as satellite imaging produces information of time-varying and spatial distribution of hydrological parameters, terrain properties, and land use. Radar (**R**adio **d**etection and **r**anging) data has been used widely in rainfall forecasting and has led to the increase in lead-time of flood forecasting. Furthermore, in real-time flood forecasting systems, it is commonplace to have a reliable and automated gauging network.

(ii) A hydrological flood forecasting model with an appropriate lead-time and proper accuracy of flood forecasts

Both the lead-time and the accuracy of forecasts are important factors to identify the quality of a forecast. Note that a detailed understanding of the physical processes is not necessarily a concern, as long as accurate forecasts are obtained. This consideration is significant since data driven models, and other classes of black-box models, are especially suited in this respect. The better the flood forecast, the better

will be the ability to mitigate flood hazard in terms of lives saved and damages prevented. However, it is not easy to achieve effective forecasts. This is because longer lead-time forecasts may often require more types of data and perhaps more complicated models. These requirements may increase forecast uncertainty, thus, decrease the accuracy of forecasts. The trade-off between model complexity and data requirements necessitates the choice of appropriate models depending on the data available. Feldman (1994), in Table 1, has summarized these points relating real-time measured data, types of model, and forecast lead-time.

Table 1.1: The properties of various types of real-time flood forecasting system (Feldman, 1994)

Type	System description	Forecast lead time	Complexity	Expense	Uncertainty
1	Real-time measurement of stream flow with routing/correlation to key locations	Shortest	Simplest	Least	Least
2	Same as 1 but including precipitation in correlation to key locations	Short	Simple	Little	Little
3	Real-time measurement of precipitation, streamflow, evaporation, etc plus a catchment runoff model	Long	More complex to very complex	More to very	More
4	Same as 3 but including a Meteorological model	Longer	Very complex	Very	More
5	Same as 4 but including weather forecasts	Longest	Most complex	Most	Most

In some case studies, even a simple flood forecasting system using only flood routing components can have a sufficient lead-time of forecasts if the forecast locations are sufficiently far downstream from flood-producing events. In the case

where the flood routing model cannot provide forecasts with sufficient lead-time, then a combination of catchment runoff models (i.e., rainfall-runoff models) and flood routing models is preferred.

(iii) Updating techniques to improve efficiency of the forecasting model

In forecasting problems a model is expected to generalize the relationship between input and output of the modeled system, and whenever inputs are given the outputs can be predicted. However, forecast errors exist since there are always uncertainties associated with data and processes which may be excluded in the model. Hence in a real-time forecasting system the use of the most recent observed data including forecast errors to improve the forecasting model is an important issue. This is often considered as updating procedures or data assimilation approaches in real-time forecasting problems.

(iv) Assessment of uncertainty involving model predictions

This issue arises since uncertainties associated with forecasts will impact the decisions taken by managers. Information on uncertainty does not make decision-making easier, but to ignore it is to ignore reality (Loucks, 2005). According to Maskey (2004), incorporating what is known about the uncertainty into input variables and parameters used in optimization and simulation models can help to quantify the uncertainty of forecasts (i.e., the outputs of the forecasting model). The uncertainty in flood forecasting consists of the uncertainty in (1) the quantity, (2) the temporal distribution over the forecast period, and (3) the spatial distribution of rainfalls over the catchment and lateral flow into rivers as well. Therefore, the ability to model and quantify uncertainty is important but not an easy task in a real-time flood forecasting system. The reliability of expected outcomes depends on many factors. They are usually split into a number of sources of uncertainty such as meteorological component that involves rainfall estimation, and hydrologic-hydraulic components that involve rainfall-runoff modelling. These sources of uncertainties are assumed to meet Bayesian assumptions and can be determined individually. More details about uncertainty analysis for hydrological modelling is presented in Krzysztofowicz (2001), where appropriate hydrologic uncertainty processors (HUP) and meteorological uncertainty processors (MUP) are introduced.

(v) Computation capability of a computer center for processing data and running forecasting model

Having a sufficiently high computation capability is an essential requirement for a flood forecasting system since for real-time purposes, the forecasting model requires substantial levels of computation within a limited time. Nowadays, powerful computers and efficient computing techniques (e.g., parallel computing, grid computing) have been applied to satisfy this requirement.

(vi) Human factors in the operation of a real-time flood forecasting system

Operators who operate real-time flood forecasting systems also play an important role. The efficiency of this system depends on the experience and skill of operators. At a workshop by the International Institute for Applied Systems Analysis (1982), experts from many countries agreed that no real-time operation of forecasting systems or controlling hydro-systems should ever be made without a competent person supervising it. Thus, training operators with the proper skills is an important requirement in a real-time flood forecasting system as well.

The present study focuses on a hydrological flood forecasting model because it acts as the “heart” of a flood forecasting system and thus focuses on points (ii) and (iii) mentioned above. There is a wide range of hydrological models which have been developed and applied for the flood forecasting problem. However, no model can claim absolute superiority. Hence, it is important to examine the relative advantages and disadvantages of the selected model, or modeling paradigm, with respect to factors such as data availability, hydrological factors affecting the river system under study, and ease of setting up the model. However, it should be realized that it is not realistic to expect a flood forecasting model to issue accurate forecasts all the time, since a model is a simplified representation of the natural flood processes. Hence, it is necessary to improve the accuracy of original model results by updating model output and/or parameters. This has been discussed in (iii) above and will also be addressed in this thesis.

1.3. Aims and Objectives of the Study

The aim of this research is to investigate the application of data driven modeling for flood forecasting in large river systems, with the Lower Mekong taken as a case study. The Adaptive Neuron-Fuzzy System or ANFIS is chosen for this purpose. The model would be sufficiently flexible to be trained from observed data to represent hydrological relationships of river flood processes. An effective learning algorithm would train the model by using measured data in order to generalize the relationships between inputs (e.g., rainfalls and/or upstream discharges) and outputs (e.g., discharges and/or water levels at forecasting locations) in a river system. An adaptive real-time flood forecasting model also means that when used on-line, the model is able to use observed data to provide on-line forecast and provide 'feedback' information (i.e., the deviations from forecasting results and observed ones) efficiently to improve its forecast on-line, thus enhancing real-time flood forecast accuracy.

The objectives of the research include:

- (a) To review various approaches for real-time flood forecasting in the literature. The review will focus on neural and fuzzy approaches since they are the important computational intelligence approaches that enable one to develop hybrid models for non-linear and on-line adaptive applications. The review begins with neural approaches since much of the learning capabilities in current state-of-the-art neural fuzzy models comprise of the learning techniques found in this approach.
- (b) To evaluate the performance of the Adaptive Neuron-Fuzzy Inference System (ANFIS) for the flood forecasting problem of the Lower Mekong Basin based on the benchmark used by Mekong River Commission and other performance indexes. A number of gauging stations along the Lower Mekong River will be chosen as case studies. In addition, a comparison between results given by the Adaptive Neuron-Fuzzy Inference System and the results obtained by the model currently used at the Flood Management and Mitigation Centre of Mekong River Commission would be presented. Model performance will be studied in relation to the use of various combinations of inputs including

water level and average rainfall for selected stations on the Lower Mekong River.

- (c) To evaluate the application of a local learning ANFIS model for water level forecasts.
- (d) To elucidate the advantages and limitations of ANFIS for flood forecasting applications in the Lower Mekong.

1.4. Structure of the Report

Chapter 2 provides a literature review of a Real-Time Flood Forecasting System. The major components of this system and related issues are described. The most recent and commonly applied hydrological modelling techniques suitable for flood forecasting are presented. Specifically, the Unified Run-off Basin Simulation (URBS), an operational model currently in use by the Regional Flood Management and Mitigation Center (RFMMC) of the Mekong River Commission (MRC) for flood forecasting is described in detail. In Chapter 3, the theoretical background of ANFIS is presented. First is the concept of Adaptive Network which is a general Feed-Forward Neural Network with a standard supervised learning technique based on the gradient descent method. Second is the concept of hybrid learning technique which uses the Least Squares Estimate method to indentify linear parameters while using gradient descent method to identify non-linear parameters. In addition, the on-line learning technique is described. This method allows model parameters to be adjusted based on feed-back information (i.e., recent output error) when the model is in operation. Third is the concept of Fuzzy logic, Fuzzy rules and Fuzzy Inference Systems. And finally, ANFIS will be presented. This is a hybrid technique that combines the advantages of Neural Networks (i.e., learning capability) and the Fuzzy Inference System (i.e., dealing with fuzzy concept). Chapter 3 also reviews real-time updating techniques which attempts to use feedback information (i.e., recent output error) to either correct model output or model parameters, thus, to improve model performance. In addition, a review of flood forecasting models which have been used to deal with flood forecasting problem for the Lower Mekong is presented. In Chapter 4, the hydrological characteristics of the Lower Mekong

River and flood events in the past are described. This chapter provides key information for a number of gauging stations where water levels were forecasted by ANFIS and describes the data collection, and methodology for ANFIS model development for water level forecasts. In Chapter 5, the results, analysis and discussions on ANFIS performance for the Lower Mekong are presented. This chapter shows a comparison between ANFIS, with and without the implementation of an output error correction procedure, with URBS. The material in the Chapter 5 has been accepted for publication in *Hydrological Processes* journal (Nguyen and Chua, 2011b). Chapter 6 investigates the link between hydrologic factors and model performance. River flows at various gauging stations along a large river can differ in terms of the flow contributions from upstream stations, rainfall and lateral flows from sub-basins immediately upstream of the station of interest. Therefore, Chapter 6 provides an analysis on how these factors influence ANFIS model performance by developing a series of ANFIS models for various stations along the Lower Mekong. The material in this chapter has been submitted for possible journal publication. Chapter 7 presents the results of ANFIS with on-line learning with real-time data. The study in Chapter 7 demonstrates the application of ANFIS models trained with an on-line learning algorithm. Chapter 8 provides conclusions obtained from the present study.

CHAPTER 2

LITERATURE REVIEW

2.1. Introduction

Flood forecasting methods can be categorized into various types depending on the methodology used of forecasting model. The common types of river flood forecasting methods mentioned in the literature are as follows.

- i) Physically-based method: this method utilizes physically-based/deterministic models that are on the mathematical equations of mass, momentum and energy conservation principles to describe hydrological processes of river basins and channels. Conceptual models such as the TANK model (Sugawara, 1995) are a class of physically based models that obey the mass conservation principle, while more sophisticated distributed models are based on both mass and momentum conservation principles and in general require more information.
- ii) Statistical method: this method is based on statistical time-series analysis such as autoregressive moving average (ARMA) or autoregressive moving average with exogenous inputs (ARMAX), introduced by Box and Jenkins (1976). ARMA and ARMAX models have been used to forecast water level or discharge in rivers (Abrahart and See, 2000). However, these linear regression models do not represent the highly nonlinear dynamics inherent in flooding processes well and therefore may not always perform adequately (Hsu *et al.*, 1995).
- iii) Data-driven method: this method utilizes the data-driven modelling approach that is in order to model an input-output relationship of hydrological processes by a direct input-output mapping constructed from observed input-output samples. The computational intelligence tools such as Artificial Neural Network (ANN), Neuro-Fuzzy Inference Systems (NFIS), Support Vector Machines (SVM) are common data-driven tools for hydrological forecasting applications. These tools employ statistical learning algorithms to learn or approximate the non-linear input-output relationship from given training samples.

- iv) Probabilistic method: this method quantifies uncertainties in terms of a probability distribution function associated with the forecasts provided by a hydrological deterministic model. Krzysztofowic (1999) presented the fundamentals of utilizing a Bayesian forecasting system to produce a probabilistic forecast via a deterministic catchment model. Krzysztofowic (2002a and 2002b) demonstrated the use of probabilistic methods for flood forecasting problems. The author suggested that the probabilistic flood forecasting method can help the authorities not only to issue flood warnings with explicitly stated probabilities, but also to establish risk-based criteria for flood warnings and emergency responses.
- v) Ensemble method: the main idea of this method is to use a number of models running in parallel to provide various forecasts of river water level at each time step. These forecasts are used to infer a probability distribution function of the forecast at each time step. Georgakakos et al. (2004) demonstrated the use of multimodel ensembles to characterize streamflow. The ensemble method is computationally expensive and thus requires performed in high performance computing system.

The present study focuses on data-driven method for the river flood forecasting problem as data-driven model is cost effective to set up than a physically-based model and it can archive acceptable accuracy in real-time flood forecasting. In order to clarify the difference between physically-based and data-driven methods for flood forecasting, the principles of these contrasting methods are reviewed. The applicability, advantages and disadvantages of these approaches which have been found in previous studies are also summarized. Data assimilation which is a technique to improve accuracy of real-time forecast is presented as well. In addition, this chapter highlight the challenges of flood forecasting in large rivers. The more detailed reviews pertinent to specific topics studied are presented in later chapters.

2.2. Physically-Based Modelling Approach

Physically-based modeling approaches aim to analyze and provide insight into systems based on knowledge of their physical properties and principles. These principles are usually represented in mathematical forms (i.e., partial differential equations of conservation laws for quantities such as mass, momentum, energy, etc). These laws enables one to model the evolution of the systems in both the time and space domains. Hence a physically-based model can be integrated by many sub-processes, distributed in both the time and space. Physically-based models represent the ‘‘physics’’ as they are best understood (Freeze and Harlan, 1969). For most problems, a numerical solution is sought by discretizing time-space dimensions into a discrete set of nodes. This implies that such models require that data on the physical characteristics of the watershed such as evaporation rates, soil type, soil moisture, slope of the soil surface, land use and management situations in the catchment area, etc, be available at the model grid scale. It has to be mentioned that such data is sometimes not available, even in heavily instrumented research watersheds. In addition, the more detailed the physics to be simulated, the more complex the model and the more data and computational time needed. Physically-based models can be further divided, in increasing levels of complexity, as conceptual, lumped and fully distributed models. An example of a fully distributed model is the MIKE SHE model, developed by the Danish Hydraulic Institute (DHI). It is an integrated distributed hydrological model which includes all the major processes in the land phase of the hydrologic cycle. MIKE SHE has been used as a tool for the analysis, planning and management of a wide range of water resources and environmental problems related to surface water and groundwater.

In order to model hydrodynamic processes of water flow there are many models varying from one-dimensional models (e.g., MIKE 11, HEC-RAS), two-dimensional models (e.g., MIKE21, SOBEK) to three-dimensional models (e.g., MIKE 3, Delft3D). An important issue in physically-based modeling of stream flow is that the model performance is strongly dependent on model structure. Increasing model complexity does not always lead to better simulations because of limitations such as the model structure itself, information available for calibration, accuracy

and representation of rainfall, and the parameter estimation procedure employed (Butts et al, 2004). The selection of an appropriate model structure should take into account accuracy, computational time, and data availability. The advantage of physically-based modeling approaches is that it allows one to simulate a system completely and model changes in the behavior of the system under various scenarios. However, there is a trade-off between model complexity and data requirements. Thus, the drawback of a fully distributed model is that the model usually requires an extensive database to accurately represent the system, and in many applications, such data are often not available or contain high levels of uncertainties. In addition, modeling a complicated system with complex interactions between many components usually requires much data and extensive computational time, more than is feasible for real-time applications.

In real-time applications, conceptual models are often preferred. The conceptual model is an attempt to take only the most essential sub-processes of a phenomenon into account within a lumped structure and parameters. The main advantage of a conceptual model for real-time forecasting purpose is that the time for running the model is very fast and satisfies the strict time restriction. A conceptual model does not attempt to model a system to such a detail as a fully distributed physically-based model, but it extracts from the system only the most essential principles at a conceptual level (i.e., some complicated processes and complex properties of the system are lumped as simple components and coefficients respectively). Thus, a conceptual model can reduce significantly the complexity of modeling in comparison with a fully distributed physically-based model. Hence it provides two main advantages: it is easy to understand, and it requires much less computational time for simulation and is therefore feasible for real-time applications for a large and complicated system. However, since processes and characteristics of the system are lumped in the conceptual model, this class of model cannot simulate the evolution of the system in such detail in both time and space domains as a fully distributed model.

In the context of river flow modeling, conceptual models are developed for rainfall-runoff and flow routing as well. The rainfall-runoff process is usually modeled by a lumped conceptual model. This type of model considers a catchment

as the combination of a number of water storages (i.e., surface, soil moisture, groundwater storages). Each of the water storages and the interaction between them are characterized by a number of state variables representing the amount of water in these storages and several lumped conceptual coefficients. A lumped conceptual rainfall-runoff model takes rainfall as the model input which changes the values of state variables of the model and then produces the runoff (i.e., the outflow) at the outlet of the modeled catchment. This process is illustrated in Part 1 of Fig. 2.1. Part 2 in Fig 2.1 is the component for the flow routing process. It receives outflow given by the rainfall-runoff model as the input and models the flow at key locations as the output. For hydrologic river flow routing (i.e., conceptual model for river routing), the Muskingum method, developed by McCarthy (1938), and which is based on continuity equation and a storage relationship that depends on both inflow and outflow is commonly used.

Figure 2.1: The typical frame work for stream flow simulation with two parts: (1) Rainfall-Runoff and (2)- Flow routing.

The conventional physically-based approach for river flow forecasting is the incorporation of a conceptual model for rainfall-runoff and a one-dimensional hydrodynamic river flow model. HEC-RAS (Hydrologic Engineering Center's River Analysis System, developed by the US Army Corps of Engineers

(www.hec.usace.army.mil/software/hecras) and MIKE FLOOD developed by the Danish Hydraulic Institute (www.mikebydhi.com) are common software that adopt this approach. In this approach, the whole catchment is first divided into sub-catchments according to the catchment's topographic characteristic after which the rainfall-runoff conceptual models are used to model each sub-catchment. The discharges at outlets of the sub-catchments are used as inputs for the one-dimensional hydrodynamic model. In this model the river system is characterized by cross-sections along the river and their roughness coefficients. The flood plains can be simplified as wide channels that are connected to the river via weirs. The initial and boundary conditions need to be defined or given. Upstream boundaries are time series of discharges given by conceptual rainfall-runoff models representing the sub-catchments. Downstream boundaries often are discharge and/or water level time series at downstream boundary locations. In the simulation phase, one must first set up a model of the river system with appropriate structures and then calibrate the model off-line. The data collected from past flow events are used to calibrate and verify the model. In the forecasting phase the calibrated model is used in the on-line mode in which real-time hydrological data is applied to the model to estimate future stages or flows at cross-sections along the river. However, a well calibrated model does not necessarily mean it can give good forecasts, due to the uncertainty associated with the inputs, initial conditions, boundary conditions, and model parameters. Therefore, to improve the performance of the model in real-time flood forecasting, an updating procedure is often used.

The role of updating procedures to improve model performance in real-time flood forecasting has been reviewed by the World Meteorological Organization (WMO, 1990b). A total of 14 models from 11 countries were compared using four standard data sets. The results showed that an updating procedure can have a major impact in real-time hydrological forecasting applications. According to Askew (1991) "there is no doubt that, when used in practice, all hydrological models are associated with some form of updating procedure and any description of a model is incomplete without a description of the updating procedure used with it". Therefore, updating techniques or "data assimilation" have received much attention and are considered as essential measures needed to improve the performance of physically-

based models for real-time river flood forecasting. A detailed review of updating techniques is presented in Chapter 4. It is realized that there are many factors such as the spatial and time variability of inputs, initial conditions, and boundary conditions which can affect the real-time flood forecast given by a physically-based or conceptual model considerably. Therefore it is reasonable to incorporate an updating procedure to update output variables, state-variables and/or model parameters of the physically-based or the conceptual model to improve forecasts.

2.2.1. The Unified Run-off Basin Simulation (URBS) Model

The conceptual runoff routing model Unified Run-off Basin Simulation (URBS) (Carroll, 2007) is currently implemented by the RFMMC for operational forecasts.

The URBS has been used extensively for flood forecasting by the Australian Bureau of Meteorology (BoM) and also by the Changjiang Water Resources Commission (CWRC) in China. URBS is a hydrologic modeling program that enables the simulation of catchment storage and runoff response by a network of conceptual storages representing the stream network and reservoirs. It combines two hydrological modeling processes into one model: (i) A rainfall runoff model, which converts the gross rainfall into net or excess rainfall; and (ii) A runoff routing model, which takes the excess rainfall as input and converts it into flow.

For runoff-routing modelling, URBS allows the setup of a runoff-routing networked model of sub-catchments based on centroidal inflows. Two runoff routing models are available to describe catchment and channel storage routing behaviour. These are the URBS Basic and Split routing models. The Basic model assumes that the catchment and channel storage for each sub-catchment is lumped together and represented as a single non-linear reservoir. Each conceptual non-linear reservoir is represented by the storage-discharge (S-Q) relationship:

$$S = k_c Q^m = \frac{\alpha f L n (1 + F)^2}{\sqrt{S_c} (1 + U)^2} Q^m \quad (2.1)$$

where S is catchment and channel storage, α is store lag parameter, f is reach length factor, L is length of reach, U is fraction urbanisation of sub-catchment, F is fraction of forested sub-catchment, n is channel roughness (Manning), S_c is channel slope, Q is outflow, m is catchment non-linearity parameter and k_c is non-linear routing constant. For the Split model, catchment and channel routing in each sub-catchment are separated. Taking into consideration the effect of slope on velocity, the time taken for runoff to travel from sub-catchment perimeter to its centre is given.

$$T = \frac{\sqrt{\frac{A}{\pi}}}{v} \quad (2.2)$$

where A is area of sub-catchment and v is velocity of flow. The runoff is then routed through a non-linear catchment:

$$S_{catchment} = \frac{\beta \sqrt{A} (1 + F)^2}{(1 + U)^2} Q^m \quad (2.3)$$

where $S_{catchment}$ is catchment storage, and β is catchment lag parameter. For channel routing, the non-linear Muskingum model is adopted in which lag time is assumed to be proportional to the reach length (Carroll, 2007):

$$S_{channel} = \alpha f \frac{nL}{\sqrt{S_c}} [xQ_u + (1 - x)Q_d]^p \quad (2.4)$$

where $S_{channel}$ is channel storage, α is channel lag parameter, f is reach length factor, L is length of reach, S_c is channel slope, Q_u is inflow at upstream end of reach, Q_d is outflow at downstream end of channel reach, x is Muskingum translation parameter, p is Muskingum non-linearity parameter and n is channel roughness.

For Rainfall-runoff modelling URBS adopts two rainfall loss approaches; event based and continuous modelling. Event based modelling requires the user to specify the rainfall lost to the catchment before surface runoff occurs. This loss is commonly called the initial loss. Various loss models can then be used to estimate rainfall losses during the event. The URBS model models rainfall losses for impervious and pervious areas separately. The user however has the option of specifying loss parameters that represent both impervious and pervious areas if so required. URBS has two methods that can be used to include the rainfall excess of continuous rainfall runoff modelling. The user can select to use URBS' simple initial loss model or can opt to integrate the results from a third party water model such as the Australian Water Balance Model or AWBM (Boughton, 2004) model. More details on rainfall-runoff modeling in URBS can be found in Carroll (2007).

In order to develop a URBS model for a large river basin for the flood forecasting problem, a modeler may have to do a number of steps briefly described as follows: (i) divide the entire basin into sub-basins by using a GIS tool such as CatchmentSIM (<http://www.csse.com.au/>) which allows the use of a DEM (Digital Elevation Model) as input and provides boundaries and sub-basin areas. (ii) Set up and calibrate rainfall-runoff URBS models for sub-basins. (iii) Connect the sub-models and carry out flow routing to route the runoff to a gauging station. (iv) Convert discharge to water level at interested gauging stations using rating curves.

The Mekong River Basin is divided into sub-basins and modeled by forty nine URBS sub-models which are linked together by three channel routing models i.e., from Chiang Saen to Kratie, the Tonle Sap system and from Kratie to Tan Chau/Chau Doc (MRC, 2009). The models were calibrated using daily historical hydrological data in the MRC's hydrological database. During operation, rainfall forecasts are obtained from the daily Satellite Rainfall Estimates (SRE) provided by the National Oceanic and Atmospheric Administration (NOAA). SRE is a grid based product of 24 hour rainfalls derived from satellite estimates of the cloud top temperatures. The data is calibrated by NOAA to a limited number of ground truth observations obtained through the World Meteorological Organization Global Telecommunication System; this number varying from 5 to 35 stations on any particular day.

2.3. Statistical Method with Multiple Linear Regression Model

In general, a multiple linear regression model with k regressor variables has the form:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_k x_k + \varepsilon \quad (2.5)$$

where the parameters $\beta_j, j = 0, 1, 2, \dots, k$, are called the regression coefficients and ε is a random error term. This model describes a hyperplane in the k dimensional space of the regressor variables $\{x_j\}$. The parameter β_j represents the expected change in response Y per unit change in x_j when all the remaining regressors x_i ($i \neq j$) are held constant. Multiple linear regression models are often used as approximating functions. That is, the true functional relationship between Y and $x_1, x_2, \dots, x_k$ is unknown, but over certain ranges of the independent variables the linear regression model may be an adequate approximation. A model that is more complex in structure than Eqn. 2.5 may often still be analyzed by multiple linear regression techniques. For example, consider a second-order polynomial model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{33} x_3^2 + \varepsilon \quad (2.6)$$

If we let $x_4 = x_1 x_2$ and $\beta_4 = \beta_{12}$; $x_5 = x_1 x_3$ and $\beta_5 = \beta_{13}$; $x_6 = x_2 x_3$ and $\beta_6 = \beta_{23}$; $x_7 = x_1^2$ and $\beta_7 = \beta_{11}$; $x_8 = x_2^2$ and $\beta_8 = \beta_{22}$; $x_9 = x_3^2$ and $\beta_9 = \beta_{33}$ then Eqn. 2.6 can be written as

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \beta_8 x_8 + \beta_9 x_9 + \varepsilon \quad (2.7)$$

which is a multiple linear regression model with nine regressor variables. The surface of response variable Y generated by the model in Eqn. 2.6 is not linear, although it can be considered as a linear regression model under the form presented

by Eqn. 2.7. Therefore, in general, any regression model that is linear in parameters (the β 's) can be considered as a linear regression model (regardless of the shape of the surface that it generates), while a non-linear regression model must have parameters which are in non-linear relationship with each other.

Suppose that $n > k$ observations are available, and let x_{ij} denote the i th observation or level of variable x_j . The observations are

$$(x_{i1}, x_{i2}, x_{i3}, \dots, x_{ik}, y_i) \quad i=1, 2, \dots, n \text{ and } n > k$$

An appropriate regression model for n given observations with the general form presented in Eqn. 2.5 may be the one which has least sum of squares errors

$$L = \sum_{i=1}^n \varepsilon^2 = \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^k \beta_j x_{ij} \right)^2 \quad (2.8)$$

The method of least squares which minimize the sum of squares errors L can be used to estimate the regression coefficients β_j , $j = 0, 1, 2, \dots, k$.

The regression model in Eqn. 2.5 relates k regressors $x_1, x_2, x_3, \dots, x_k$ and response y , thus, the n observations can be written by n equations

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_k x_{ik} + \varepsilon_i \quad i = 1, 2, 3, \dots, n \quad (2.9)$$

Using matrix notation as follows

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix} \quad y = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} \quad \text{and} \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Those n equation of observations become as

$$y = X\beta + \varepsilon \quad (2.10)$$

where y is an $(n \times 1)$ vector of the observations, X is an $(n \times p)$ matrix of the levels of the independent variables, β is a $(p \times 1)$ vector of the regression coefficients, and ε is a $(n \times 1)$ vector of random errors. We wish to find the vector of least squares estimators, $\hat{\beta}$, that minimizes

$$L = \sum_{i=1}^n \varepsilon_i^2 = \varepsilon \varepsilon' = (y - X\beta)'(y - X\beta) \quad (2.11)$$

The least squares estimator $\hat{\beta}$ is the solution for β in the equations

$$\frac{\partial L}{\partial \beta} = 0 \quad (2.12)$$

that is

$$\hat{\beta} = (X'X)^{-1} X'y \quad (2.13)$$

The more details on Multiple Linear Regression can be found in Douglas C. Montgomery and George C. Runger (2002).

2.4. Data-Driven Modelling Approach

Data-driven modeling aims to directly model the input-output relationship between water level (or discharge) and other variables that is more often based on statistical correlations rather than on physical principles. Data driven modeling is based on modeling the response of a system to a specific input to develop a relation (or function) that relates input and output variables. It is referred as “data driven” since its parameters are identified by a training process using a set of input-output data samples. This process optimizes model parameters by minimizing the error between computed and observed output values. A data-driven model is usually quickly and easily developed and implemented. This approach is thus very useful for real-time river flow forecasting where the objective is to obtain accurate water level forecasts at specific locations in a timely fashion. In the last two decades, computational intelligence techniques such as Artificial Neural Networks (ANNs), Fuzzy

Inference Systems (FIS), Neuro-Fuzzy Inference Systems (NFIS) have been proven in many fields to be able to simulate complex, non-linear systems since they have the ability of “learning” from data.

ANNs are inspired by the human learning ability that works by a process of changing connections between neurons inside the human brain. The Multi-Layer Perceptron neural network (MLP) is the most widely used in ANN applications. This is a multiple layer feedforward neural network which consists of two or more single-layers connected in a cascade form, where the output of the preceding layer is used as the input of the following layer. Each neuron or perceptron employs a transfer function (e.g. the step function, the liner function or the sigmoid function) to transmit its input signals (a single scalar and/or a weighted sum of output signals from neurons in a previous layer through connections) to its output signal. The transfer function of the neuron multiplies the strength of input signals by a specific degree to generate the output of this neuron. A diagram of the cascade connection is shown in Figure 2.3 below:

Figure 2.2: The architecture of a typical feed-forward multi-layer neural Network

A MLP neural network is typically trained by a supervised learning method such as the back-propagation algorithm. In this algorithm, when the input of a sample is applied to the network input layer the signal would be computed and transmits

forward through the network to the output layer where the output is compared with the desired target, and an output error is computed. This error is propagated backward through the network to each node, and correspondingly the connection weights are adjusted in order to descent gradient of the network error function, therefore, minimize network output error. The forward and backward procedures are repeated sample by sample until the terminate criteria is met. The more details on MLP neural network can be found in Haykin, 1994.

A Multi-Layer Perceptron neural network is a universal approximator since it can approximate any underlying non-linear relationship (White, et al 1989). It is known that a MLP network with three layers (i.e., one input layer, one hidden layer and one output layer) and sufficient hidden neurons can approximate any highly nonlinear relationships in practice. They have been used in many applications in hydrology (Maier and Dandy, 2000; ASCE 2000a and 2000b) including river flood forecasting problem where water levels or discharges at a location of a river is forecasted based on rainfall and/or water levels, discharges at other locations (Half et al. 1993, Zhu et al., 1994, Minns and Hall, 1996). In order to compare the performance of ANNs with conventional models had been usually used before, Hsu et al (1995) showed that a MLP neural network was superior than the linear ARMAX time series model and the conceptual SACSMA model (Sacramento Soil Moisture Accounting Model) in both during calibration and validation. MLP networks have been used widely in flood forecasting studies for various river systems such as in Dhaka, Bangladesh (Liong et al., 1996), the Blue Nile River, Ethiopia, (Antar et. al., 2006; Shamseldin, 2010), the Yangtze River, China (Dawson et al., 2002) and the Huai River, China (Solomatine and Xue, 2004).

Fuzzy Inference Systems (FIS) are qualitative modeling schemes inspired by the human reasoning ability where the system behavior is described using a natural language (Takagi and Sugeno, 1985). Recently, the ANN and FIS have been integrated into a single framework referred to as Neuro-Fuzzy Inference Systems (NFIS). One such NFIS is the Adaptive Neuro-Fuzzy Inference System or ANFIS proposed by Jang (1993). ANFIS has been developed for rainfall-runoff and flood modelling with promising results (Nayak *et al.*, 2005; Chau *et al.*, 2005; Chen *et al.*, 2006). FIS and NFIS are described more detail in Chapter 3.

The Support Vector Machine (SVM) is another tool for data-driven modelling. It is a learning technique based on the concept of support vectors and the Structure Risk Minimization principle (Vapnik, 1998). The support vectors are the closest vectors to an optimal hyperplane that separates a data set of vectors into two different classes with maximal margin. They, thus, define the optimal hyperplane. The Structure Risk Minimization (SRM) principle suggests model identification in data-driven modelling should achieve the optimal relationship between the amount of available data, the quality of approximation and the value that characterizes the complexity of a model's structure. According to Vapnik (1998), the SRM principle is useful for statistical learning in data-driven modelling when data is limited. A description of SVM for regression or so-called Support Vector Regression (SVR) is presented here as it can be used for forecasting applications. Given n pairs of observed samples, $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, a linear approximation function for the data has a general form of $y = (w \cdot x + b)$, where w and b are parameter vectors. For a conventional linear regression model, the error/residual for the approximation of the data sample (x_i, y_i) is defined as $e_i = y_i - (w \cdot x_i + b)$. However, for a linear SVR model, the error is considerable if the absolute difference between the observed value y_i and computed value $(w \cdot x_i + b)$ is larger than ε only. Therefore, the typical definition of error in conventional linear regression model, $e_i = y_i - (w \cdot x_i + b)$, is replaced by a so-called Vapnik's ε -insensitive loss function $L_\varepsilon(y_i)$

$$L_\varepsilon(y_i) = \begin{cases} 0 & \text{for } |y_i - (w \cdot x_i + b)| \leq \varepsilon \\ |y_i - (w \cdot x_i + b)| - \varepsilon & \text{for } |y_i - (w \cdot x_i + b)| \geq \varepsilon \end{cases} \quad (2.14)$$

where ε is the error tolerance parameter. The learning algorithm for SVR model is based on the SRM principle where a tradeoff between the quality of the approximation and the complexity of the approximating function is taken into consideration. Hence the learning process minimizes the total loss and at the same time it minimizes the magnitude of vector parameter w to obtain a smooth function. Linear functions may be inadequate to approximate the inherit nonlinearity of a system in real-world problems, thus, it is more reasonable to apply nonlinear SVR

model for these problems. The main idea of non-linear SVR is firstly to transform the input vector in a higher dimensional feature space, in which the data may exhibit linearity, and then to perform as linear SVR in this feature space. Support Vector Machines are able to approximate nonlinear regression functions even when the data is corrupted by noise (Vapnik, 1998) and they outperform conventional regression models (Drucker et al., 1997). According to Liong and Sivapragasam (2002), SVMs have a number of advantages over ANNs such as (i) better generalization when the training data set is small, (ii) improving time series analysis, and (iii) no local minima problems. The authors also reported that SVM outperformed ANN in a flood stage forecasting application for Dahaka in Bangladesh. Yu et al. (2006) demonstrated a good performance of SVR for stage forecasting of Lan-Yang River in Taiwan. Yu et al. (2004) showed that SVM was able to incorporate chaos theory in order to analyse chaotic time series. In addition, the SVM parameters could be optimized using evolutionary algorithm. The author reported this combined approach provided advantages over the conventional approach using ARIMA in their case studies of forecasting runoff time-series of the Tryggevælde catchment, Denmark and the Mississippi River at Vicksburg. Choy and Chen (2003) demonstrated the use of SVM to determine objectively the structure of the radial basis function networks thus they could optimize the network to model the rainfall-runoff relationship for the Fuji River in Japan. Therefore, SVM together ANN, NFIS can be considered as data-driven models alternative to physically-based models for hydrologic forecasting problems.

However, data-driven method has its own shortcoming. A data-driven model may not achieve success in an application if the quantity and quality of data available are inadequate. Where data is limited, training data-driven model based on the Empirical Risk Minimization principle may lead to overfitting problem because when training data set is small, the Empirical Risk Minimization principle can achieve good fits with the given data; however, the model may fit the noise in the data as well. Therefore, its generalization becomes poor and may lead to poor performance during testing. Vapnik (1998) suggested that the Structure Risk Minimization (SRM) principle should be applied for model training instead of the Empirical Risk Minimization principle when the training data set is small. Data-

driven models has are able to perform well when tested within the range of training data set, however, may not perform well when tested outside the range of training data. Extrapolation outside the range of training data thus may not provide guarantee good results from the model. The further the extrapolation outside the range of the training data, the less accurate is expected in the model forecast.

In a data-driven model, the model should be trained properly to avoid both underfitting and overfitting problems. If a selected model's structure and or a number of iterations in training is inadequate, then the model may have underfitting problem where both training and testing errors are not acceptable. In contrast, if a model is trained to very fit the training data well, it may fit the noise and its generalization may be poor, leading to over-fitting problem. In order to avoid these problems, two issues are considered: (i) to select an adequate model structure (e.g. the order of a polynomial regression model, or a number of hidden layers and numbers of hidden neurons in hidden layers of a ANN, or a number of rules in a neuro-fuzzy inference system, etc.) and (ii) to train the model optimally before testing errors increase. In practice, this can be accomplished using a validation process. However, these methods may not be effective if the available data set is small since a small data set may be inadequate to present the input-output relationship of the modeled system. To deal with the problem of small available date set in data-driven modelling, a learning method such as the Structure Risk Minimization (SRM) principle (Vapnik, 1998) can provide a solution.

2.5. Data Assimilation

In a forecasting system, the accuracy of forecasts depends not only on the accuracy of input information but also the accuracy of the estimate of model state variables. Data assimilation is a technique that allows the computation of the probability density function of the model state variables conditioned on the measured observations allowing the probability density function the model solution to be obtained. Therefore, we can address the most likely model solution within the lower and upper bounds which represent the uncertainty of the solution. It is expensive to represent accurately the full probability density function for high dimensional

simulation models. Therefore, a data assimilation technique must usually represent the probability density function using statistical moments or an ensemble of model states and then searches for estimators such as the mean and maximum likelihood with the associated covariance representing the uncertainty of variables. Basically, data assimilation techniques are based on Bayesian theory where one can improve the posterior probability distribution function of a variable by conditioning on the available observations of the related variables. The standard Kalman filter (KF) proposed by Kalman (1960) can be used for sequential data assimilation for linear dynamics and measurement processes with Gaussian error statistics (Gelb, 1974). The extended Kalman filter (EKF) was developed for nonlinear dynamics systems, although it is notoriously unstable if the nonlinearities are strong (Miller *et al.* 1994). Both the KF and the EKF explicitly propagate error information with a dynamic equation for the state error covariance matrix. However, the integration of this equation is not computationally feasible for high-dimensional systems. The ensemble Kalman filter (EnKF) was proposed by Evensen (1994) in order to overcome this problem. The EnKF is based on ideas of EKF, and in addition it uses a collection of state vectors, called an ensemble, to represent the distribution of the system state and replace the covariance matrix by the sample covariance computed from the ensemble. Reichle *et al.* (2002) demonstrated the application of the ensemble Kalman filter to the estimation of soil moisture from L-band microwave brightness temperature observations. The author reported that with relatively few ensemble members the EnKF yields reasonable soil moisture estimates. Madsen and Skotner (2005) proposed a combined filtering and error forecasting procedure to improve the performance of an operational flood forecasting, implemented in the MIKE 11 flood forecasting system. This procedure was demonstrated in an operational flood forecasting setup for Metro Manila, Philippines. Gill *et al.* (2007) demonstrated a SVM model for soil moisture prediction coupled with EnKF by which subsequent observations and their statistics were assimilated to update predictions from the SVM model. The authors concluded that the SVM-EnKF algorithm performed well and improved the model predictions considerably.

2.6. Data Driven Models - Challenges Faced for Large Rivers

As shown in the previous section, modeling a large river using a physically-based approach requires dividing the whole large basin into many sub-basins. Much time and experience is required to calibrate the sub-models for those sub-basins. In addition, when these sub-models are integrated, the accumulation of errors from sub-models can cause large errors in the final results. The data-driven modeling approach in comparison is quick to develop since the primary function is not to model the actual physical processes with time-spatial variation properties. For a large river basin the amount of data needs to be prepared and processed is very large and the computational resources required may be high. In contrast, data driven models do not have such high computational demands.

Studies carried out on flood forecasting with data driven models are usually focused on model performance and investigations into the link between the flow behavior and input requirements for data driven models in forecasting applications are rare. In addition, most forecasting studies are carried out for short term forecasts (one or two time steps ahead). In order to assess the utility of data driven models, however; more studies focusing on the accuracy of data driven models in long term forecasting need to be carried out. In addition, the hydrological conditions can play a significant role in determining the inputs that are significant. To have a good definition of the hydrological conditions, requires many gauging stations and measurements at short time intervals. However, in practice the number of gauging stations is usually limited and data collected infrequently. For example, rainfall data is an important input but is usually estimated since rain gauges are sparsely distributed within the river basin. In addition, data collection and transmission can also be problematic due to the lack of resources.

CHAPTER 3

METHODOLOGY

3.1. Introduction

Artificial Neural Network (ANN) and Fuzzy Inference System (FIS) are two well-known techniques in Computational Intelligence which have been used widely in applications of non-linear system modelling and control. Recently, researchers in Computational Intelligence have developed a more advanced technique which is a hybrid system combining both the concepts of ANN and FIS and called a neuro-fuzzy system. A common framework of a neuro-fuzzy system is to represent FIS in ANN architecture and utilize the learning capability of ANN to optimize the FIS parameters. The Adaptive Neuro-Fuzzy Inference System (ANFIS) proposed by Jang (1993) is a well-know approach to neuro-fuzzy systems.

The adaptive network is a superset of all kinds of feed-forward neural networks with supervised learning capability (Jang, 1993). An adaptive network, as its name implies, is a network structure consisting of nodes and directional links through which the nodes are connected. Moreover, part or all of the nodes are adaptive, which means their outputs depend on the parameter(s) pertaining to these nodes, and the learning rule specifies how these parameters should be changed to minimize a prescribed error measure. It is considered as a superset of feed-forward neural networks since each node can perform a particular function (node function) on incoming signals as well as a set of parameters pertaining to this node. The formulas for the node functions may vary from node to node, and the choice of each node function depends on the overall input-output function which the adaptive network is required to carry out. For example, a common Multi-layer Perceptron Neural Network is an adaptive network with sigmoid functions for hidden nodes and linear function for output nodes. The Radial Basis Function Neural Network is also an adaptive network with radial basis functions for hidden nodes and linear functions for output nodes.

Jang (1993) proposed an adaptive network with appropriate functions for nodes in its layers to represent the Tagaki-Sugeno Fuzzy Inference System which has the capability of modeling nonlinear relationships when an appropriate structure and parameters are identified. In a framework of an adaptive network, the Tagaki-Sugeno Fuzzy Inference System can be used to learn from data. This approach combines the learning capability of the Artificial Neural Network and reasoning capability of the Fuzzy Inference System to model highly nonlinear processes.

3.2. Adaptive Network with Structure and Basic Learning Rule

An adaptive network (see Fig. 3.1) is a multilayer feed-forward network in which each node performs a particular function (node function) on incoming signals as well as a set of parameters pertaining to this node. The formulas for the node functions may vary from node to node, and the choice of each node function depends on the overall input-output function which the adaptive network is required to carry out. Note that the links in an adaptive network only indicate the flow direction of signals between nodes; no weights are associated with the links.

Figure.3.1. An adaptive network with two inputs and two outputs.

In Fig.3.1, the circle and square nodes in an adaptive network reflect that different nodes may have different adaptive capabilities. A square node (adaptive node) has parameters while a circle node (fixed node) has none. The parameter set

of an adaptive network is the union of the parameter sets of each adaptive node. In order to achieve a desired input-output mapping, these parameters are updated according to given training data and a gradient-based learning procedure described below.

Suppose that a given adaptive network has L layers and the k th layer has $\#(k)$ nodes. We can denote the node in the i^{th} position of the k^{th} layer by (k,i) , and its node function (or node output) by O_i^k . Since a node output depends on its incoming signals and its parameter set, we have

$$O_i^k = O_i^k(O_1^{k-1}, \dots, O_{\#(k-1)}^{k-1}, a, b, c, \dots) \quad (3.1)$$

where a, b, c , etc., are the parameters pertaining to this node. Note that O_i^k denotes the output for both the output node and node function. Assuming that the given training data set has P entries, the error measure for the p^{th} ($1 \leq p \leq P$) training data entry can be represented as the sum of squared errors:

$$E_p = \sum_{m=1}^{\#(L)} (T_{m,p} - O_{m,p}^L)^2 \quad (3.2)$$

where T_m is the m^{th} component of p^{th} target output vector, and $O_{m,p}^L$ is the m^{th} component of actual output vector produced by the presentation of the p^{th} input vector. Hence the overall error measure is $E = \sum_{p=1}^P E_p$. The purpose of a learning

process from the given training data set is to optimize model parameters in order to minimize the overall error E . The basic learning rule for this purpose is based on gradient descent and the chain rule, proposed by Werbos (1974). After the publication by Werbos (1974), Rumelhart et al (1986) developed a standard learning method for the well-known Multiple-Layer-Perceptron Neural Network. Jang (1993) used the formula in Rumelhart *et. al.* (1986) to present the basic learning rule for an adaptive network. In order to develop a learning procedure that

---

implements gradient descent in E over a parameter space, first the error rate $\partial E_p / \partial O$ for the p^{th} training data and for each node output O is calculated. The error rate for the output node at (L,i) can be calculated readily from (3.2):

$$\frac{\partial E_p}{\partial O_{i,p}^L} = -2(T_{i,p} - O_{i,p}^L) \quad (3.3)$$

For the internal node at (k,i) , the error rate can be derived by the chain rule:

$$\frac{\partial E_p}{\partial O_{i,p}^k} = \sum_{m=1}^{\#(k+1)} \frac{\partial E_p}{\partial O_{m,p}^{k+1}} \frac{\partial O_{m,p}^{k+1}}{\partial O_{i,p}^k} \quad (3.4)$$

where $1 \leq k \leq L - 1$. If α is a parameter of the given adaptive network, then the derivative of the error of p^{th} entry training data respect to α is defined as

$$\frac{\partial E_p}{\partial \alpha} = \sum_{O^* \in S} \frac{\partial E_p}{\partial O^*} \frac{\partial O^*}{\partial \alpha} \quad (3.5)$$

where α is the set of nodes whose outputs depend on α . The derivative of the overall error measure E with respect to α is

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^P \frac{\partial E_p}{\partial \alpha} \quad (3.6)$$

Accordingly, the update formula for the generic parameter α based on gradient descent is

$$\alpha_{\text{new}} = \alpha_{\text{old}} + \Delta \alpha \quad (3.7)$$

where $\Delta \alpha = -\eta \frac{\partial E}{\partial \alpha}$ and η is a learning rate which can be further expressed as

$$\eta = \frac{k}{\sqrt{\sum_{\alpha} \left(\frac{\partial E}{\partial \alpha} \right)^2}} \quad (3.8)$$

k is the step size, the length of each gradient transition in the parameter space. The value of k can be changed to vary the speed of convergence.

The gradient method is commonly used to identify the parameters in an adaptive network and its implementation is considered as a standard method nowadays. However, the method is slow and likely to become trapped in local minima. Jang (1993) proposed a hybrid learning rule which combines the gradient method and the least squares estimate (LSE) which is efficient when there are linear parameters in the adaptive network. The above learning algorithm is referred to as batch learning (or offline-line learning) since the update formula for parameter α is based on Eqn. 3.7 where the derivative of the overall error measure E with respect to α is computed from the whole training data set and the update action takes place only after each epoch or sweep through the entire training dataset has taken place. Besides the batch learning approach, another learning paradigm where the parameters are updated immediately after each input-output pair has been presented is referred to as pattern learning, local learning or on-line learning. This learning paradigm is vital to the on-line parameter identification for systems with rapidly changing characteristics. The latter procedure is a faster hybrid learning rule and has also been proposed by Jang (1993).

3.3. Hybrid Learning with Batch (Off-Line) Mode

The hybrid learning rule proposed by Jang (1993) combines the gradient method and the least squares estimate (LSE) to identify the parameters of an adaptive network. It is assumed that the parameters of an adaptive network can be decomposed into two sets S_1 and S_2 , and if S_1 is given then the output of the adaptive network is a linear function in the elements of S_2 . Hence, the linear parameters can be identified by the least squares estimate method. When S_2 is identified the parameters in S_1 can be updated by the gradient method. Using these

two methods, the hybrid learning procedure is faster than the standard procedure which implements the gradient method only.

When values of elements of S_1 are given, and training data is fed to the adaptive network where only a vector whose elements are parameters in S_2 are unknown. Denoting this vector as X , we can obtain a matrix equation:

$$AX=B \quad (3.9)$$

where A is a matrix of combination between given parameters and input values, B is a matrix of output values. If the number of elements in S_2 is M , the number of training samples (input-output pairs in training data set) is P and, for simplicity, assume that the adaptive network under consideration has only one output, then the dimensions of A , X and B are $P \times M$, $M \times 1$ and $P \times 1$, respectively. Since P (number of training data pairs) is usually greater than M (number of linear parameters), this is an over-determined problem and generally there is no exact solution to Eqn. 3.9. Instead, a least squares estimate (LSE) of X , X^* , is sought to minimize the squared error $\|AX - B\|^2$. The most well-known formula for X^* uses the pseudo-inverse of X :

$$X^* = (A^T A)^{-1} A^T B \quad (3.10)$$

where A^T is the transpose of A , and $(A^T A)^{-1} A^T$ is the pseudo-inverse of A if $A^T A$ is non-singular. While Eqn. 3.10 is concise in notation, it is expensive in computation when dealing with matrix inverse and moreover, it becomes ill defined if $A^T A$ is singular. As a result, the sequential formulas to compute the LSE of X can be applied instead. This sequential method of LSE is more efficient (especially when M is small) and can be easily modified in an on-line application (see below) for systems with changing characteristics. Specifically, let the i th row vector of matrix

A and the i th element of B defined in Eqn. 3.9 be a_i^T and b_i^T , respectively, then X can be calculated iteratively using the sequential formulas:

$$\left. \begin{aligned} X_{i+1} &= X_i + S_{i+1}a_{i+1}(b_{i+1}^T - a_{i+1}^T X_i) \\ S_{i+1} &= S_i - \frac{S_i a_{i+1} a_{i+1}^T S_i}{1 + a_{i+1}^T S_i a_{i+1}}, \quad i = 0, 1, 2, \dots, P-1 \end{aligned} \right\} \quad (3.11)$$

where S_i is often called the covariance matrix and the least squares estimate X^* is equal to X_P . The initial conditions to bootstrap Eqn. 3.11 are $X_0 = 0$ and $S_0 = \gamma I$, where γ is a positive large number and I is the identity matrix of dimension $M \times M$. When dealing with multi-output adaptive networks, Eqn. 3.11 still applies except that b_i^T is the i th row of matrix B .

In order to combine the gradient method and the least squares estimate to update the parameters in an adaptive network, each epoch of this hybrid learning procedure is composed of a forward pass and a backward pass. In the forward pass, when input data is fed to the network, functional signals go forward to calculate each node's output until the matrices A and B in Eqn. 3.9 are obtained, and the parameters in S_2 are identified by the sequential least squares formula in Eqn. 3.11. After identifying parameters in S_2 , the functional signals keep going forward till the error measure is calculated. In the backward pass, the error rates (the derivative of the error measure w.r.t. each node output, see Eqn. 3.3 and Eqn. 3.4) propagate from the output end toward the input end, and the parameters in S_1 are updated by the gradient method in Eqn. 3.7.

For given fixed values of parameters in S_1 , the parameters in S_2 thus found are guaranteed to be the global optimum point in the S_2 parameter space due to the choice of the squared error measure. Not only can this hybrid learning rule decrease the dimension of the search space in the gradient method, in general, it will also cut down substantially the convergence time.

3.4. Hybrid Learning with Pattern (On-Line) Mode

If the parameters are updated after each data presentation, we have the pattern learning or on-line learning paradigm. This learning paradigm is vital to the on-line parameter identification for systems with changing characteristics. To modify the batch learning rule to the on-line version, the gradient descent should be based on E_p which is the model output error corresponding to the current or p^{th} sample (see Eqn. 3.5) instead of E which is sum of errors of all samples in the given training dataset. The update formula for the generic parameter α based on gradient descent is

$$\alpha_{\text{new}} = \alpha_{\text{old}} + \Delta\alpha \quad \text{where} \quad \Delta\alpha = -\eta \frac{\partial E_p}{\partial \alpha} \quad (3.12a)$$

Strictly speaking, this is not a truly gradient search procedure to minimize E ; however, it will approximate to gradient search if the learning rate is small.

For the sequential least squares formula to account for the time-varying characteristics of the incoming data, we need to decay the effects of old data pairs as new data pairs become available. Again, this problem is well studied in the adaptive control and system identification literature and a number of solutions are presented in Goodwin and Sin (1984). One simple method is to formulate the squared error measure as a weighted version that gives higher weighting factors to more recent data pairs. This amounts to the addition of a forgetting factor λ to the original sequential formula:

$$\left. \begin{aligned} X_{i+1} &= X_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T X_i) \\ S_{i+1} &= \frac{1}{\lambda} \left[S_i - \frac{S_i a_{i+1} a_{i+1}^T S_i}{\lambda + a_{i+1}^T S_i a_{i+1}} \right] \end{aligned} \right\} \quad (3.12b)$$

where the value of λ is between 0 and 1. The smaller the value of λ , the faster the decay of older information. A small value of λ can cause numerical instability and should be avoided.

3.5. Fuzzy logic, Fuzzy rules and Fuzzy Inference Systems

Fuzzy systems are currently being used in many applications such as automatic control, pattern recognition, decision support, etc. They are systems using fuzzy logic which is based on the essential concept of fuzzy sets proposed by Zadeh (1965). The most unique characteristic of this theory, in contrast to classical mathematics, is its operation on various membership functions (MF) instead of the crisp real values of the variables. This heuristic permits fuzzy theory to be a powerful tool whenever it handles imprecise data or ambiguous non-linear relationships between the variables. Fuzzy logic appears to be effective at handling dynamic, non-linear and noisy data, especially when the underlying physical relationships are not fully understood. Since the past decade there were some initial applications using fuzzy logic approach in hydrological modelling (Fujita *et al.*, 1992; Zhu and Fujita, 1994; Zhu *et al.*, 1994; See and Openshaw, 1999; Stuber *et al.*, 2000; Hundecha *et al.*, 2001) which demonstrates that fuzzy systems can be applied efficiently in hydrological engineering applications.

3.5.1. Fuzzy Logic

Fuzzy logic starts with the concept of a fuzzy set. A *fuzzy set* is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership. This is an appropriate way to represent vague or fuzzy concepts (e.g., fast runner, hot weather, tall men, etc.) which are very common in linguistics which form the basis of human reasoning. Fuzziness is used to describe uncertainty related to event ambiguity, that is, the subjective degree to which an event occurs.

A *membership function* (MF) is a function that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is usually referred to as the *universe of discourse*. A fuzzy set is an extension of a classical set. If X is the universe of discourse and its elements are denoted by x , then a fuzzy set A in X is defined as a set of ordered pairs: $A = \{x, \mu_A(x) \mid x \in X\}$ where $\mu_A(x)$ is called the membership function (or MF) of x in A . The membership function maps each element of X to a membership value

between 0 and 1. Triangular, Bell shape and Gaussian functions are common types of membership functions (Fig. 3.2):

- Triangular membership $\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (3.13a)$

- Bell-shape membership $\mu(x) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (3.13b)$

- Gaussian membership $\mu(x) = \exp \left[-\frac{(x-c)^2}{\sigma^2} \right] \quad (3.13c)$

Figure 3.2. Examples of membership functions

Fuzzy logic forms the basis for human communication and human reasoning. It is an appropriate tool to express knowledge of domain experts (i.e., in terms of IF-THEN rules) so that human experience can be incorporated into modeling complex nonlinear systems. Fuzzy logic can model nonlinear functions of arbitrary complexity and it makes a system more flexible and tolerant of imprecise data. Therefore, fuzzy logic is powerful tool for dealing quickly and efficiently with imprecision and nonlinearity.

3.5.2. Fuzzy IF-THEN Rules

Fuzzy if-then rules or fuzzy conditional statements are expressions of the form IF A THEN B, where A and B are labels of fuzzy sets (Zadeh, 1965) characterized by appropriate membership functions. Due to their concise form, fuzzy if-then rules are often employed to capture the imprecise modes of reasoning that play an essential role in the human ability to make decisions in an environment of uncertainty and imprecision. In an IF-THEN fuzzy rule, the IF part is called the antecedent or premise and the THEN part is called the consequent. An example that describes a relationship between rainfall in an area and discharge of a drainage channel is “*IF rainfall intensity is high and rainfall duration is long, THEN discharge is much*” where rainfall intensity, rainfall duration and discharge are linguistic variables, a *high* value of rainfall intensity, *long* time of rainfall duration and *much* value of discharge are linguistic values or labels that are characterized by membership functions.

Another form of fuzzy if-then rule, proposed by Takagi and Sugeno (1983), has fuzzy sets involved only in the premise part. By using Takagi and Sugeno’s fuzzy if-then rule, the relation between velocity and the velocity head can be described as follows: “*IF velocity is high, THEN head = $k*(velocity)^2$* ” where, high in the premise part is a linguistic label characterized by an appropriate membership function. However, the consequent part is described by a nonfuzzy equation of the input variable, velocity.

Both types of fuzzy if-then rules have been used extensively in both modeling and control. Through the use of linguistic labels and membership functions, a fuzzy

IF-THEN rule can easily capture the spirit of a “rule of thumb” used by humans. From another perspective, due to the qualifiers on the premise parts, each fuzzy IF-THEN rule can be viewed as a local description of the system under consideration. Fuzzy IF-THEN rules form a core part of the fuzzy inference systems.

3.5.3. Fuzzy Inference Systems

The aim of a fuzzy inference system is to use a known fuzzy relationship to infer from known input to unknown output where the fuzzy relationship are formed by fuzzy rules in the general form: *if* (a set of condition are satisfied) *then* (a set of consequences can be inferred). For example, there is a given fuzzy rule such as “*IF rainfall intensity is high and rainfall duration is long, THEN discharge is much*”. Now the given input which rainfall intensity is ‘moderately high’ and rainfall duration is long then a fuzzy inference system has to infer the magnitude of the discharge. The typical Fuzzy Inference System (FIS) consists of five functional blocks as shown in Fig.3.3.

Figure 3.3. A diagram of a Fuzzy Inference System (Jang, 1993)

The five functional blocks in Fig.3.3 are:

- a *rule base* containing a number of fuzzy if-then rules;

- a *database* which defines the membership functions of the fuzzy sets used in the fuzzy rules;
- a *decision-making* unit which performs the inference operation on the rules;
- a *fuzzification interface* which transforms the crisp inputs into degrees of match with linguistic values;
- a *defuzzification interface* which transform the fuzzy results of the inference into a crisp output.

Usually, the rule base and the database are jointly referred to as the *knowledge base*. The steps of fuzzy reasoning (inference operations upon fuzzy if-then rules) performed by fuzzy inference systems are as follows: (i) Compare the input variables with the membership functions on the premise part to obtain the membership values (or compatibility measures) of each linguistic label. (This step is called fuzzification step). (ii) Combine (usually by multiplication or min operator) the membership values on the premise part to get the firing strength (weight) of each rule. (iii) Generate the qualified consequent (either fuzzy or crisp) of each rule depending on the firing strength. (iv) Aggregate the qualified consequents to produce a crisp output (defuzzification.)

In order to identify a Fuzzy Inference System one of the essential tasks is to divide input and output spaces into fuzzy domains. Other tasks include defining membership functions and addressing fuzzy rules. These tasks can either be done by experience or by training from data (e.g., neural-fuzzy systems) or a combination of both techniques. Identifying a fuzzy rule based on expert knowledge is usually subjective, thus, the approach that uses input-output training data to identify a fuzzy inference system has been developed based on the Takagi - Sugeno (TSK) fuzzy inference system. In this system the fuzzy rules has fuzzy sets involved only in the premise part, whereas, the consequent part is a crisp function. The functions in the TSK can be any kind of functions of the crisp input variables. However, the common TSK fuzzy system is the first order TSK (TSK1) in which linear functions are used in consequents of fuzzy rules. For example, the fuzzy rule i (R_i) has the general form:

IF (u_1 is A_{i1}) AND (u_2 is A_{i2}) AND... AND (u_p is A_{ip}) THEN $y_i = f_i(\underline{u})$

where $f_i(\underline{u}) = w_{i0} + w_{i1}u_1 + w_{i2}u_2 + \dots + w_{ip}u_p$ is a linear function in the consequent, and A_{ij} are membership functions of fuzzy sets according to input u_j ($j = 1, 2, \dots, p$) of the rule R_i . An example of TSK shown in Fig 3.4 is a TSK fuzzy inference system with two fuzzy rules in which x and y are variables and the output of each rule is a linear function:

(i) Rule 1: IF x is A_1 and y is B_1 THEN $f_1 = p_1x + q_1y + r_1$,

(ii) Rule 2: IF x is A_2 and y is B_2 THEN $f_2 = p_2x + q_2y + r_2$.

When a pair of input values is applied to this fuzzy inference system it yields a firing strength for each rule such as w_1 for Rule 1 and w_2 for Rule 2. For this example the *min* operator is used to perform the “and” fuzzy logic operator in the inference process. Other examples include the *algebraic product*. The overall output f is the weighted sum of each rule’s output. Jang (1993) proposed a well-known approach which uses a Feed-Forward Neural Network to represent the TSK fuzzy inference system. This network can be trained by input-output training data. Therefore, the TSK fuzzy inference system will be identified by a learning process from such that data.

Figure 3.4: An example of Tagaki-Sugeno-Kang fuzzy inference

3.6. ANFIS: Adaptive Neuro-Fuzzy Inference System

ANFIS is based on the Tagaki-Sugeno-Kang fuzzy inference system embedded within the structure of the ANN (Jang, 1993). Fig.3.5 shows a schematic of ANFIS with 2 rules for a system consisting of 2 inputs and 1 output, such as:

$$(i) \text{ If } x \text{ is } A_1 \text{ and } y \text{ is } B_1, \text{ then } f_1 = p_1x + q_1y + r_1, \quad (3.14)$$

$$(ii) \text{ If } x \text{ is } A_2 \text{ and } y \text{ is } B_2, \text{ then } f_2 = p_2x + q_2y + r_2.$$

This ANFIS model, based on the TSK fuzzy inference system is shown in Fig 3.4.

Figure 3.5. A schematic of ANFIS with 2 rules for a system consisting of 2 inputs and 1 output according to Jang (1993).

The function of each layer in ANFIS is briefly described as follows:

Layer 1: Every node i in this layer is a square node (an adaptive node) with a node function

$$O_i^1 = \mu_{A_i}(x) \quad (3.15)$$

where x is the input to node i , and A_i is the linguistic label (small, large, etc.) associated with this node function. In other words, O_i^1 is the membership function (denoted by MF) of A_i , and it specifies the degree to which the given x satisfies the quantifier A_i . $\mu_{A_i}(x)$ is usually chosen to be bell-shaped with maximum equal to 1 and minimum equal to 0, such as:

$$\mu_{A_i}(x) = \frac{1}{1 + \left| \frac{x - c_i}{a_i} \right|^{2b_i}} \quad (3.16)$$

where $\{a_i, b_i, c_i\}$ is the parameter set. As the values of these parameters change, the bell-shaped functions vary accordingly, thus exhibiting changes in the membership functions on the linguistic label A_i . In fact, any continuous and piecewise differentiable functions, such as commonly used trapezoidal or triangular-shaped membership functions, are also candidates for node functions in this layer. Parameters in this layer are referred to as premise parameters.

Layer 2: Every node in this layer is a circle node (fixed node) labeled Π which multiplies the incoming signals and sends the product out. For instance:

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad i = 1, 2. \quad (3.17)$$

Each node output represents the firing strength of a rule.

Layer 3: Every node in this layer is a circle node (fixed node) labeled N . The i^{th} node calculates the ratio of the i^{th} rule's firing strength to the sum of the firing strengths all possible rules:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2. \quad (3.18)$$

For convenience, outputs of this layer will be called normalized firing strengths.

Layer 4: Every node i in this layer is a square node (adaptive node) with a node function:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (3.19)$$

where $\bar{w}_i$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer will be referred to as consequent parameters.

Layer 5: The single node in this layer is a circle node (fixed node) labeled Σ that computes the overall output as the summation of all incoming signals:

$$O_1^5 = \text{overall output} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{w_i} \quad (3.20)$$

The learning algorithm in ANFIS optimizes the ANFIS parameters that include the premise parameters, which describe the shape of the membership functions, and the consequent parameters, which describe the overall output of the system.

More precisely, the output f in Fig.3.5 can be rewritten as:

$$\begin{aligned} f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 = \bar{w}_1 f_1 + \bar{w}_2 f_2 \\ &= (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2 \end{aligned} \quad (3.21)$$

If premise parameters $\{a_1, b_1, c_1, a_2, b_2, c_2\}$ and values of input variables are given, then f is linear in the consequent parameters $\{p_1, q_1, r_1, p_2, q_2, r_2\}$. Therefore, the hybrid learning algorithm presented Sections 3.3 and 3.4 can be applied directly. More specifically, in the forward pass of the hybrid learning algorithm, functional signals go forward till Layer 4 and the consequent parameters are identified by the least squares estimate. In the backward pass, the error rates propagate backward and the premise parameters are updated by gradient descent. The Fuzzy Logic Toolbox

of MATLAB (2008), which implements the off-line training version of ANFIS (Section 3.3), was used in this study.. The online implementation of ANFIS in the present study has been developed based on Jang's code published in (<http://www.cs.nthu.edu.tw/~jang/publication.htm>) written in C programming language. Details of the hybrid on-line learning algorithm are presented in Section 3.4.

3.7. Real-Time Updating Techniques

Due to the natural complexities associated with the numerical model, temporal-spatial variation of the processes being modeled and watershed properties, predictions made by flood forecasting models will contain errors. A forecast error is the difference between the true measurement and the computed value given by the forecasting model for a certain lead time. To improve forecast accuracy, the forecasting model can be updated when the most recent data becomes available. WMO (1992) reviewed many hydrological models used for real-time forecasting applications and concluded that no model was superior for all the cases considered, and recommended that updating techniques were necessary to improve the forecasts. According to WMO (1992), among the four common updating schemes: model input variables, model parameters, state variables and model output variables updating, the model output updating is the most widely used as this approach can be applied to any simulation model as the method is independent of structure.

In order to correct output results from a model, in general, linear regressive time-series models such as the Auto-Regression or AR (Serban and Askew, 1991; Xiong and O'Connor, 2002) or Auto-Regression with moving Average or ARMA (Broersen, 2007) models have been utilized. For this purpose, an AR or ARMA model is calibrated off-line using the forecasting model output error time series, and subsequently used to predict the output errors of the forecasting model in real-time. The predicted errors are then simply added to the original forecasts to give the updated model output. The success of the AR updating procedure is dependent mainly on the degree of serial correlation in the forecast error time series of the substantive model (Serban and Askew, 1991; Shamseldin, 1996). In a further study,

Shamseldin and O'Connor (1999) used the linear Auto-Regression Model with Exogenous Inputs (ARX) model for output updating. In the ARX updating procedure, the original model output time series produced by the substantive forecasting model constitutes the exogenous input, which is used together with the available observed values in providing the updated forecast. Shamseldin and O'Connor (2001) and Shamseldin (2010) used a neural network model which can be considered as a form of a Non-Linear Auto-Regression Model with Exogenous Inputs (NARX-ANN) to replace the ARX in output updating.

The AR model is non-adaptable since after calibration its parameters are fixed. If the AR model is used for output error prediction, then it may not be appropriate if the statistical characteristics of the output error time-series are changing. Hence, the Recursive Auto-Regression (RAR) model is introduced as promising model for output error prediction since its parameters can be adjusted by a recursive least-squares algorithm that places higher weighting to the most recent errors so as to obtain a more accurate prediction at the next time step.

3.7.1 Recursive Auto-regression Models

The auto-regression (AR) model is a stochastic model that is useful in the representation of a stationary time series (Box and Jenkins, 1976). In this model, the current value y_t of the process is expressed as a finite, linear aggregate of previous values $y_{t-1}, y_{t-2}, \dots, y_{t-p}$ of the process and a random error a_t as follows:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + a_t \quad (3.22)$$

where $\phi_1, \phi_2, \dots, \phi_p$ are autoregressive parameters and p is the order. a_t is the value at time t of a pure 'white noise' sequence with zero mean and constant variance, σ_a^2 . By taking the expectation $E[a_t] = 0$, the estimate of y_t is given by:

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} \quad (3.23)$$

AR model parameters can be estimated using the least-squares approach which minimizes the standard sum of squared forward-prediction errors.

A drawback of the AR model is that it is non-adaptable since it uses offline calibration. This means then when applying an AR model in online mode, its parameters cannot be adjusted even if there is a considerable change in the character of the error time series. In contrast, the recursive auto-regression (RAR) model is adaptable as its parameters can be adjusted by a recursive least-squares algorithm that places emphasis on the most recent errors. A general recursive estimation algorithm is given by the following equations:

$$\begin{aligned} \hat{\theta}_t &= \hat{\theta}_{t-1} + K_t(y_t - \hat{y}_t) \\ \hat{y}_t &= \psi_t^T \hat{\theta}_{t-1} \\ K_t &= Q_t \psi_t \end{aligned} \quad (3.24)$$

where $\hat{\theta}_t$ is the parameter estimate at time t , y_t is the observed output at time t and $\hat{y}_t$ is the prediction of y_t based on observations up to time $t-1$. $\psi_t^T = (y_{t-1}, y_{t-2}, \dots, y_{t-p})^T$ is the regression vector that consists of previous values of measured outputs, where p is the order of the recursive regressive model. The gain, K_t , determines how much the current prediction error $(y_t - \hat{y}_t)$ affects the update of the parameter estimate. The algorithm tries to estimate a set of model parameters by which the least-squares errors at specific number of the previous time steps is obtained. These parameters, therefore, are assumed appropriate for the current time step to yield a prediction for the output variable at the next time step. The present study uses the RAR model with the Forgetting Factor algorithm implemented in the System Identification Toolbox, MATLAB (2008). This algorithm is described mathematically as:

$$\hat{\theta}_t = \hat{\theta}_{t-1} + K_t(y_t - \hat{y}_t) \quad (3.25a)$$

$$\hat{y}_t = \psi_t^T \hat{\theta}_{t-1} \quad (3.25b)$$

$$K_t = Q_t \psi_t \quad (3.25c)$$

$$Q_t = \frac{P_{t-1}}{\lambda + \psi_t^T P_{t-1} \psi_t} \quad (3.25d)$$

$$P_t = \frac{1}{\lambda} \left(P_{t-1} - \frac{P_{t-1} \psi_t \psi_t^T P_{t-1}}{\lambda + \psi_t^T P_{t-1} \psi_t} \right) \quad (3.25e)$$

where P_t is the scaled covariance matrix of the parameters at time step t . In operational use, at the initial step, P is usually initiated by a default matrix P_0 that is 10^4 times the identity matrix of order p which is the order of the recursive regressive model. At each time step t , P_t is estimated by the Eqn. 3.25e in order to minimize the function $\sum_{k=1}^t \lambda^{t-k} e_k^2$ which is a weighted sum of squared errors in the k previous time steps, where $e_k = (y_k - \hat{y}_k)$ ($k = 1, 2, \dots, t$) is the prediction error at the k^{th} time step, and λ is the forgetting factor. This weighted sum of squared errors discounts the old errors exponentially such that at the current step t , the weight equals 1, while at the k^{th} time step the weight equals λ^{t-k} .. The error at a step older than $\tau = \frac{1}{1-\lambda}$ typically carries a weight less than 0.3, thus τ represents the memory horizon of the algorithm. Typically, λ has a positive value between 0.970 and 0.995 (MATLAB user guide, 2008). Further details of the recursive estimation algorithm (i.e. Eqn. 3.24 and 3.25) can be found in Ljung (1987).

CHAPTER 4

DATA USED

4.1. Floods in the Lower Mekong River

The Mekong is the tenth-largest river in the world. The Mekong River drains a total land area of 795,000 km² from the eastern watershed of the Tibetan Plateau to the Mekong Delta in Cambodia and Vietnam. The Mekong River, shown in Fig 4.1(a), flows approximately 4,900 km through three provinces of China, continuing into Myanmar, Laos, Thailand, Cambodia and Vietnam before discharging into the South China Sea (www.mrcmekong.org). The Mekong can be divided into two parts: the upper part (about 2,200 km) which is from its source in China to the Golden Triangle area where the borders of Thailand, Laos, China and Myanmar come together and the lower part (2,700 km) from the Golden Triangle area to the river mouth in Vietnam (see Fig 4.1(b)). The lower Mekong is managed by governments of the Lao People's Democratic Republic (Laos), Thailand, Cambodia and Vietnam based on agreements for the sustainable development of water and related resources in the Lower Mekong Basin. For this purpose, The Mekong River Commission (MRC) was established in 1995 as an inter-governmental river basin organization established by the riparian countries. Its mission is to develop work programs and strategies for regional cooperation and basin-wide planning for an economically prosperous, socially just and environmentally sound Mekong River Basin. The Lower Mekong Basin (basin area = 600,000 km²) lies within the countries of Thailand, Laos, Cambodia and Vietnam (Fig 4.1(b)) and it is home for about 30 million people. It is both geographically and hydrologically variable with significant differences in topography, land cover, and climate. Elevation is highest in the north-east and reduces in the direction from north-east to south-west. The basin is generally flat in the Tonle Sap floodplain of Cambodia and the Mekong Delta in Vietnam.

Figure 4.1 (a) Location map for the Mekong Basin, (b) Rainfall distribution for the Lower Mekong Basin, (c) Sub-basin 1: 16, 906 (km²), (d) Sub-basin 2: 10,407 (km²), (e) Sub-basin 3: 119,707 (km²), (f) Sub-basin 4: 28,815 (km²). Gauging stations on main stream, tributaries, rainfall stations are shown in (c) – (f). Source MRC (2004), (2006).

The climate of the Lower Mekong Basin, which is almost always hot and often humid, is classified as tropical monsoonal. In the warmest months of March and April, average temperature ranges from 30°C to 38°C. Cooler temperatures prevail from November to February. At higher elevations in Laos, winter temperature averages 15°C. The distribution of mean annual rainfall over the Basin follows a distinct east-to-west gradient (www.mrcmekong.org). Fig 4.1(b) shows that most of the rainfall of the Lower Mekong is distributed in the highland regions of Laos and Cambodia in the areas east of the Mekong within the reach from Chiang Sean to Kratie. The largest sub-basin (sub-basin 3 in Fig 4.1(e)) has an area of 119,707 km² and is to the west of the Mekong, located in Thailand and has annual rainfall of 1,000 to 1,500 mm. The highest mean annual rainfall of between 2,500 mm to 3,000 mm occurs in sub-basin 1 in Fig 4.1(c) (16,906 km²). Therefore, tributaries to the east of the Mekong between Luang Prabang and Kratie contribute the main source to the flow in the mainstream of the Lower Mekong River.

Tropical typhoons moving from the east over the Gulf of Tonkin to the Lower Mekong Basin bringing with it heavy rains, is the typical cause for individual peaks of the monsoonal river flows. Heavy rains with long durations occurring during September and October when the seasonal discharge is already high, tends to generate significant peaks in water levels and discharge. Historically, these events have been responsible for many of the most extreme flood events observed within the Mekong system. For example, in year 2000, the largest floods on record occurred when typhoons caused heavy rains continuously over the entire basin. The first typhoon traversed northern Vietnam in the middle of July and brought heavy rainfall to the mountainous regions of Laos. Heavy rains produced floods in July, which kept water level at 10m elevation at Phnom Penh for a period of 10 days. This flood also caused water levels at Tonle Sap to be high (the Tonle Sap functions as a flood retarding basin and buffer). Following the first typhoon two other typhoons hit the central part of Vietnam in the end of August and the beginning of September 2000, causing significant rainfalls in a wide area in central Vietnam and the Mekong delta. This resulted in significant floods in the eastern parts of Laos and Cambodia, with water levels at Tonle Sap and Phnom Penh reaching record levels. The number of casualties was over 800 including about 340 and 440 people in

Cambodia and Vietnam, respectively (about 320 children died in the floods). In Cambodia, about 617,000 ha of rice field was inundated, 318,000 houses were damaged at a cost estimated to be about 80 million US dollars. In Vietnam, about 93,000 ha of rice field were destroyed and 865,000 homes were inundated for periods of weeks to months. The total cost of the damage was about 350 million US dollars (Kazama et.al., 2002)

4.2. Review of Flood Forecasting Models for the Lower Mekong

Before 2009, flood forecasting for the Lower Mekong was managed by a system that utilized models developed during the 1970s. The Streamflow Synthesis and Reservoir Regulation model (SSARR), developed by the US Corps of Engineers was used to model the rainfall-runoff process for the upper reaches from Chiang Saen to Pakse, while regression models were used to model the flows downstream from Pakse. A new flood forecasting system has recently been developed and put into operation at the Regional Flood Management and Mitigation Centre of the Mekong River Commission (RFMMC-MRC). This system was developed in 2007 in order to improve short and especially medium term forecasts. Flood forecasting for the Mekong River upstream of Pakse is now modeled by the Unified Run-off Basin Simulation (URBS) hydrological model (Section 2.2.1), which is a physically-based lumped parameter model. Because of the flatter terrain, a 1-D hydrodynamic model (ISIS hydraulic model, Halcrow) is used to model the flow downstream of Pakse. Thus, Pakse is the boundary between the URBS and ISIS models and the forecast at Pakse is used as the upstream boundary for the ISIS model. Therefore, the accuracy of forecasts at Pakse is important as it can significantly influence the accuracy of downstream forecasts. The Regional Flood Management and Mitigation Center (RFMMC) of the Mekong River Commission (MRC), located in Phnom Penh, currently operates the URBS model and provides official forecasts along the Lower Mekong with lead times ranging from 1 to 5 days (<http://ffw.mrcmekong.org>).

4.3. Hydro-Meteorological Data

Hydro-meteorological gauging stations in the Mekong River Basin have been setup and managed by the Mekong member states to measure daily water level, discharge and rainfall. These gauging stations were setup at various times. The water level at Stung Treng, Cambodia, was measured from as early as 1910. Rainfall at Takeo and Kompong Speu in Cambodia was measured from 1917, and measured from 1920 at Vientiane in Laos. However, most of the data was measured from the 1960s or 1970s. Recently, the Mekong River Commission constructed 30 water level and rain gauging stations between 2006 and 2009. Further details on the new gauging stations can be found in Sompong (2009)

Before 1995, the hydro-meteorological data of Mekong was managed separately by member states. At present, the data management system HYMOS from WL | Delft Hydraulics is used for archiving the hydro-meteorological data, both at the MRC Secretariat and by the riparian countries. By the end of April 2005, the MRC's hydro-meteorological data collection network comprised 849 stations located in Laos, Vietnam, Thailand and Cambodia, as summarised in Table 4.1.

Table 4.1. Summary of MRC Hydro-meteorological stations

Data	No. of stations
Water level and discharge	242
Rainfall	503
Synoptic	44
Sediment	60
Total	849

The MRC has also setup a near real-time water level and rainfall management system, summarised in Table 4.2, where data, are sent from line agencies in riparian countries to the MRC Secretariat daily by e-mail, daily during flood season (June to October) and weekly during the dry season (November to May). The data are stored in the operational database for input into regional forecasts, and is also disseminated to forecasting agencies in member countries for use in their national

forecasts. More details of MRC's hydrological data management are presented in Hedlund et.al (2005)

Table 4.2 Near real-time hydro-meteorological data managed by the MRC

Item	Flood season (Jun - Oct)	Dry season (Nov - May)
Water level Data (stations)	44	19
Rainfall Data (stations)	44	19
Data delivery to MRC Secretariat	Daily	Weekly
Forecasting activity	Flood Forecasting	River Monitoring
Forecasting point	21	19
Day-ahead forecast	5-day forecast	7-day forecast

4.3.1. Gauging Stations under Study

Figure 4.3 shows the mean annual flow at gauging stations along the Lower Mekong from Chiang Saen to the delta (with the locations shown in Fig 4.1(b)). The daily flow data from 1994 to 2003 was used to estimate the mean annual flow at these stations. As shown in Fig 4.3, the contribution from the Upper Mekong in China to the Lower Mekong as determined by the flow measured at Chiang Saen, is about 15%. The lateral flows from tributaries between Chiang Saen and Nong Khai in Sub-basin 1 increase the flow in the Lower Mekong River by a modest 12%. The major contribution to the flows in the Lower Mekong occurs between Nong Khai to Kratie where the total flow is increased from 27% to 91%. Within this reach, the main stream flow is increased significantly between Nong Khai and Thakhek (increase of 21%) and between Pakse and Kratie (increase of 33%) with major lateral flows contributed by Sub-basin 1 (16,906 km²) and Sub-basin 4 (28,815 km²), respectively. Therefore, Thakhek, Pakse and Kratie can be seen as key stations which represent major contributions to the Lower Mekong River, and are thus selected for this study. Besides water level, rainfall data measured at rain gauges within the Sub-Basins of interest were also used in this study.

Figure 4.2. The increase of the Mean Annual Flow along the Lower Mekong from Chiang Sean to the delta in term of volume and percentage.

4.4. Evaluation of Forecast Accuracy

The error indexes used to evaluate forecast accuracy in the present study are the mean absolute error (MAE):

$$MAE (m) = \frac{\sum_{i=1}^n |H_i - \hat{H}_i|}{n} \quad (4.1)$$

and the root mean square error (RMSE):

$$RMSE (m) = \sqrt{\frac{\sum_{i=1}^n (H_i - \hat{H}_i)^2}{n}} \quad (4.2)$$

where n is the total number of data elements, H_i and $\hat{H}_i$ are the observed and the predicted water levels at the i^{th} time interval respectively, are statistical measures of the average quantity of the residuals/ errors of water level forecasts. The mean

absolute error treats all errors equally while the root mean square error stresses more on large errors due to the squared terms. In addition, the coefficient of the efficiency (CE) defined as (Nash and Sutcliffe, 1970):

$$CE = 1 - \frac{\sum_{i=1}^n (H_i - \hat{H}_i)^2}{\sum_{i=1}^n (H_i - \bar{H})^2} \quad (4.3)$$

where $\bar{H}$ is the mean value of observed water levels was also used. CE can range from $-\infty$ to 1. It is a normalized statistic that determines the relative magnitude of the residual variance (“noise”) compared to the measured data variance (“information”). An efficiency of 1 ($CE = 1$) corresponds to a perfect match of the forecast and the observed water level. The closer the coefficient of efficiency is to 1, the more accurate the forecast is. An efficiency of 0 ($CE = 0$) indicates the model performance is poor since its accuracy is no better than the mean value. If the efficiency is less than zero ($CE < 0$) it indicates that the mean observed value is a better predictor than the simulated values. Sevat and Dezetter (1991) found Nash-Sutcliffe efficiency to be the best objective function for reflecting the overall fit of a hydrograph. In addition, Nash-Sutcliffe efficiency is recommended to use by ASCE (1993). Therefore, the CE has been used for evaluating model performance in the present study. Lastly, the Persistence Index (PI) (Kitanidis and Bras, 1980):

$$PI = 1 - \frac{\sum_{i=1}^n (H_i - \hat{H}_i)^2}{\sum_{i=1}^n (H_i - H_{i-L})^2} \quad (4.4)$$

where H_{i-L} is water level estimation from a so-called persistence/ naïve model that simply takes the last observed water level at time i minus the lead time L as a forecast for water level at time i is also used. A persistence index of 1 ($PI = 1$) means a perfect match of the forecast and the observed water level. The closer the PI is to 1, the more accurate is the forecast. If $PI = 0$, this indicates that the forecast model performs just as well as the persistence or naïve model. If $PI < 0$, this

indicates that the forecast model performs even worse than the persistence or naïve model.

It is typical that there will be a preponderance of water level data in the low and medium range with lesser number of data points in the high water level range because of seasonal influences. This is problematic since evaluating a model based on the entire dataset using overall model performance indices such as the root mean square error (RMSE), coefficient of efficiency (CE), and persistence index (PI) would be biased towards the low and medium water level conditions. This is not ideal as model performance at the high water levels are usually of interest. The dataset was therefore separated into low, medium, high and very high regimes as this will allow the performance of the model in the different regimes to be independently assessed. The four water level regimes at a station were defined according to the 50%, 80% and 95% percentile of the water levels measured at a given station. Thus, the low, medium, high and very high water levels (H) correspond to the water levels where $H < 50\%$, $50\% < H < 80\%$, $80\% < H < 95\%$ and $H > 95\%$, respectively. The values for low, medium, high and very high water levels at Thakhek, Pakse, and Kratie are shown in Table 4.3. In addition, flood levels and alarm levels defined by the Mekong River Commission (MRC) are included (<http://ffw.mrcmekong.org/>). According to the MRC, flood level is the level at which overbank flow occurs, and alarm level is the level where conditions are critical and water levels are close to flood level. It is observed that the alarm level roughly corresponds to the 95th percentile level of the data analyzed.

Table 4.3. Water level regimes, alarm and flood levels for Thakhek, Pakse, and Kratie

Water Level Regimes and Warning Levels	Levels (m)		
	Thakhek	Pakse	Kratie
Low regime	< 9.5	< 7.5	< 16.5
Medium regime	9.5 - 12.0	7.5 - 10.0	16.5 - 19.5
High regime	12.0 - 13.0	10.0 - 11.5	19.5 - 21.0
Very high regime	> 13.0	>11.5	> 21.0
Alarm level	13.0	11.0	22.0
Flood level	13.5	12.0	23.0

4.5. Data Normalization

The absolute values of the different data types (rainfall, discharge, level etc) used in a model can differ by orders of magnitude. The learning procedure of a data-driven model based on techniques such as neural networks, nearest-neighbor classifiers, or clustering usually use a distance-based minimization algorithm for analysis. Thus, if the data is left un-normalized, the distance measures calculated over a larger magnitude will artificially outweigh distance measurements calculated on parameters with smaller differences. Therefore, the model tends to focus subjectively on large magnitude variables and loses information given by parameters with smaller magnitudes.

Normalizing/scaling data in a specific range such as [0.0, 1.0] is, thus, usually applied in data-driven modelling. In practice, min-max normalization is commonly applied. It performs a linear transformation on the original data. Min-max normalization preserves the relationships among the original data values. It will encounter an “out-of-bounds” error if a future input case for normalization falls outside of the original data range (Han and Kember, 2006). In case of utilizing ANN models with sigmoid functions for hidden nodes, normalization must be

applied in order to restrict the data range to the interval of zero-to-one since the output of the nodes with sigmoid function can vary in this interval only. Minns and Hall (1996) reported that the choice of the range (i.e., min and max values) for normalization may influence significantly the performance of the ANN. The authors showed the ANN failed to simulate the extremes in the data sets which were out-of-range of the applied normalization. In a flood forecasting problem, a future flood event may occur with higher magnitude than what have been presented in the training data. Therefore, it is common practice to include the maximum measured value in the training set.

CHAPTER 5

THE DATA-DRIVEN APPROACH AS A FLOOD FORECASTING MODEL

5.1. Introduction

Accurate forecasting of water levels and/or discharge is a key requirement for the successful implementation of flood management and mitigation in a river basin. In most instances of real-time flood forecasting, accurate water level forecasts at critical locations or gauging station locations along a river are of concern rather than whether the model includes hydrological or hydraulic processes (Plate, 2007). The statistical approach includes time series analysis methods such as autoregressive moving average (ARMA) or ARMA with exogenous inputs, introduced by Box and Jenkins (1976). ARMA and ARMA with exogenous input models have been used to forecast water level or discharge in rivers (Abrahart and See, 2000). However, these linear regressive models do not represent the highly nonlinear dynamics inherent in flooding processes well and therefore may not always perform adequately (Hsu et al., 1995). In the last two decades, data-driven models based on computational intelligence tools such as artificial neural networks (ANNs) and fuzzy inference systems (FISs) have been adopted in hydrological modelling applications. These computational intelligence tools are able to ‘learn’ the nonlinear relationship of a given process. ANNs are inspired by the human learning ability that works by a process of changing connections between neurons inside the human brain. The ANNs therefore can learn an underlying phenomenon from its input–output samples (Dawson and Wilby, 1998; ASCE, 2000a, 2000b). FIS are qualitative modelling schemes inspired by the human reasoning ability where the system behaviour is described using a natural language (Takagi and Sugeno, 1985). There have been a growing number of applications of FIS in water resources engineering (Zhu et al., 1994; See and Openshaw, 1999, 2000; Xiong et al., 2001).

Recently, the ANN and FIS have been integrated into a single framework referred to as Neuro-Fuzzy Inference Systems (NFIS). One such NFIS is the Adaptive Neuro-Fuzzy Inference System or ANFIS proposed by Jang (1993). Nayak *et al.* (2005) used ANFIS to forecast hourly flow at the Satrana gauging station in the Kolar Basin (basin area = 904 km²) in India for 1 to 6 hours flood forecasts. The authors used 4 input variables consisting of the current discharge at Satrana and the areal average rainfall lagged at 7, 8 and 9 hours, as inputs for the ANFIS model. They found that the ANFIS model was able provide good results for longer lead times compared to the ANN model. Chen *et al.* (2006) developed ANFIS models to forecast the hourly water level at the Ziuchian Bridge gauging station at Choshui River (basin area = 3,157 km²) in central Taiwan for one and two time-steps ahead. The authors developed three different ANFIS models with three different sets of inputs: (i) three previous water levels at Ziuchian Bridge and three previous water levels at Chungyun Bridge, 26 km upstream of Ziuchian Bridge, (ii) differenced water levels at Ziuchian Bridge and Chungyun Bridge, and (iii) areal average rainfall at the 5 previous time steps in addition to the inputs used in (ii). The authors reported that (i) a downstream water level can be suitably estimated by just using several of the previous downstream and upstream water levels as input to the ANFIS model; (ii) using the differenced inputs could remove the non-stationarity of a time series and provide better performance; and (iii) including the watershed average rainfall enhanced forecasting accuracy. In addition, the authors compared a standard back-propagation neural network (BPNN) using the first set of inputs to compare with ANFIS for 1-hour ahead forecasts and found that ANFIS model performed better than the BPNN model. Chau et al (2005) compared 1-day-ahead flows forecasted by a genetic algorithm-based artificial neural network (ANN-GA) and the adaptive-network-based fuzzy inference system ANFIS at Han-Kou, situated along the Yangtze River. The authors used the water levels at Han-Kou and Luo-Shan, situated 150 km upstream as inputs. Lateral inflows were neglected since the flows were small in comparison to the discharge in the main river. The authors reported that both the ANN-GA and ANFIS models produced accurate results, with a RMSE of 0.2 m. Although NFIS promises to be a useful tool because it is a hybrid system that captures the benefits of both the ANN and FIS (Nayak et al., 2005) the

number of studies using NFIS is relatively fewer than that of ANNs for data-driven hydrological modelling (Maier et al., 2010). More studies on NFIS for real-time flood forecasting would clarify its applicability for real world problems.

This chapter, therefore, investigates the use of ANFIS as a real-time modelling tool for flood forecasting. The motivation for this study is threefold: (i) to establish advanced neuro-fuzzy models such as ANFIS is advantageous over conventional models such as Multiple Linear Regression (MLR), Multiple Layer Perceptron (MLP) Neural Network in real-time river water level forecasting, (ii) although there have been some studies using ANFIS, these studies have not compared ANFIS results against existing operational flood forecasting models. This comparison is necessary to provide a proper evaluation of the ANFIS model in a practical application. (iii) propose an output updating procedure on the basis of the recursive AR (RAR) model. The RAR model provides advancement over traditional time series models because it is adaptive, i.e., its parameters can be adjusted in real time, using information of the most recent forecast errors.

Water level data for the Mekong River at Pakse gauging station in Laos was analysed for this purpose. This gauging station is important since it is boundary between upstream and downstream models in the Lower Mekong. This study considers multistep-ahead forecasts from 1 to 5 days, which is the operational requirement of the Regional Flood Management and Mitigation Centre (RFMMC), Mekong River Commission (MRC). ANFIS model results are compared with the results obtained by the existing flood forecasting model, URBS model, managed by the RFMMC.

5.2. ANFIS Model Development

5.2.1. Data Division

The daily water level at Pakse and several upstream stations during the wet seasons from 1993 to 1998 was used for training, whereas data from 1999 to 2000 was used for validation and data from 2001 to 2003 and 2009 were used for testing. ANFIS model results were compared with the MRC's benchmark for 2001 to 2003 and

CHAPTER 5: THE DATA-DRIVEN APPROACH AS A FLOOD FORECASTING MODEL

with results from the URBS model for the wet season in 2009 only because URBS model results are available only for 2009 onwards. Table 5.1 contains details of the water level data for Pakse used in the training, validation and testing data sets and the number of days when water levels rose above the alarm water level of 11 m (last column). The alarm level of the Mekong at Pakse is 11 m, and the danger level is 12 m.

Table 5.1. Statistical properties of water levels used in the training, validation, and testing datasets. (N = number of samples, Min, Max = minimum and maximum values, respectively, S.D. = standard deviation, Cv = coefficient of variation and Cs = skewness coefficient.

Data used	Statistical parameters							Days above alarm level (days)
	Mean (m)	N (days)	Min (m)	Max (m)	S. D. (m)	C _v	C _s	
Training (1993-1998)	5.98	1,134	1.36	13.01	3.06	0.51	0.25	53
Validation (1999-2000)	7.12	380	2.56	13.34	2.57	0.36	0.31	28
Testing (2001-2003 and 2009)	6.88	693	1.45	12.7	2.83	0.41	0.11	58

5.2.2. Input–Output Selection

An ANFIS model is a multi-input–single-output system. To provide water level forecasts at Pakse for lead times from 1 to 5 days, the present study developed five individual ANFIS models: ANFIS₁, ANFIS₂, ANFIS₃, ANFIS₄ and ANFIS₅. Each ANFIS model has one output, which is the water level at a certain lead time, l ($l = 1, 2, \dots, 5$ days). A preliminary analysis is required to identify the significant input variables for these models. This is because not all of the potential input variables

will be equally informative because some may be uncorrelated, noisy or have no significant relationship with the output variable being modelled (Maier and Dandy, 2000). The time-delay (or persistence) effect is usually significant in the relationship between water levels at the current and next one or two time steps. Therefore, it should be taken into account to identify input variables for flood forecasting (Chen et al., 2006). The present study used correlation analysis to identify the appropriate lag times of water level at Pakse that can potentially be used as inputs for the ANFIS models. The autocorrelation and the partial autocorrelation of the water level time series at Pakse are plotted in Fig 5.1.

Figure 5.1. Correlogram of the water level time series at Pakse: (a) Autocorrelation and (b) Partial autocorrelation.

Figure 5.1(a) shows that the autocorrelation in the water level time series of Pakse decreases gradually, indicating long-term memory effects, and Fig 5.1(b) shows that the interval 95% CI of the partial autocorrelation is up to three time lags. This implies that an AR process with an order of 3 can be used to model the time series of water levels at Pakse. Therefore, a general input–output form of the ANFIS models can be as follows:

$$H_{Pakse}(t+l) = f\{H_{Pakse}(t), H_{Pakse}(t-1), H_{Pakse}(t-2)\} \quad (5.1)$$

where $H_{\text{Pakse}}(t)$, $H_{\text{Pakse}}(t-1)$, and $H_{\text{Pakse}}(t-2)$ are water levels at the Pakse gauging station at the current and two previous time steps, and $H_{\text{Pakse}}(t+l)$ is the water level at Pakse at l -lead-time steps ($l=1, 2, \dots, 5$) ahead.

An analysis on flows that contribute to the total flow at Pakse revealed that the flow at Savannakhet station, located 240 km upstream from Pakse, contributes 81.6%, and the rest of 18.4% is from lateral flows discharging into the reach between Savannakhet and Pakse. This suggests that water levels at Pakse are strongly influenced by upstream water levels. The crosscorrelation method was used to analyse the correlation between water levels at upstream stations and Pakse. In addition, the root mean square errors (RMSE) of the difference between the time series of water levels at Pakse and the individual upstream stations were calculated for shifts of up to 10 days. Fig 5.2(a) shows the cross-correlation between the water levels at Luang Prabang (1180 km), Vientiane (730 km), Pak Sane (545km), Thakhek (350 km) and Savannakhet (240km) and the water level at Pakse. Fig 5.1(a) shows that the water levels up to three lags at Savannakhet has the strongest correlation with the current water level at Pakse. Thus, the water levels at Savannakhet at time t , $t-1$ and $t-2$ should be added to the general input–output form of the ANFIS models. Furthermore, Fig 5.2(b) shows the minimum of RMSE between the time series of water levels of Savannakhet, Thakhek, NongKhai, Luang Prabang, Chiang Saen and Pakse are obtained when shifting these time series one, two, three, four and six time steps, respectively. This means the lag times between Pakse and those stations are 1, 2, 3, 4 and 6 days, respectively. Therefore, water level variables of Savannakhet, Thakhek, NongKhai, Luang Prabang, Chiang Saen at time $t-1$, $t-2$, $t-3$, $t-4$ and $t-6$ are also potential inputs variable to forecast the water level at Pakse at time t . However, the cross-correlations of these variables are significantly less than the cross-correlation of water levels at Savannakhet at time t , $t-1$, $t-2$. Therefore, water levels at stations from Chiang Saen to Thakhek were not added to the general form of the ANFIS models. Finally, the general input-output form of the ANFIS models for water level forecasting at Pakse in the present study is:

$$H_{Pakse}(t+l) = g\{H_{Pakse}(t), H_{Pakse}(t-1), H_{Pakse}(t-2), H_{Sav}(t), H_{Sav}(t-1), H_{Sav}(t-2)\} \quad (5.2)$$

Figure 5.2. The relationship between the water levels at Pakse and upstream stations: (a) cross-correlation and (b) RMSE based on shifted time series

5.2.3. Model Structure Selection and Calibration

The ANFIS Editor Graphical User Interface (GUI) is a tool in the Fuzzy Logic Toolbox in MATLAB (2008) which provides an interface for users to generate, train and test ANFIS models in the batch (off-line) mode. When a training dataset is given, and a number of membership functions for each input is provide by modeler, the ANFIS Editor GUI allows the modeler generate an ANFIS structure and initial parameters which are ready for training. Each input variable, which varies within a range, is clustered into several class values to identify initial parameters for membership functions of the fuzzy rules, such as the centers of Gaussian or Bell-shape membership functions are defined by the center of the classes. There are two methods for generating an ANFIS model. They are based on two partitioning

techniques for the training dataset: grid partition and subtractive clustering. The subtractive clustering method partitions the data into clusters and generates an Fuzzy Inference System with the minimum number of rules to model the interested system, i.e., the number of rules is equal to the number of membership functions for each input. The grid partition method divides input domain based on a grid specified by the number of membership functions for inputs. The number of ANFIS rules generated when applying the grid partition method is in the order of number of inputs powers the number of membership functions for each input. Therefore, for a same modeled system, the ANFIS generated when applying subtractive clustering has much less model parameters than that generated when applying the grid partition method. If n , m , p are the number of inputs, number of membership functions, and number of parameters for a particular membership function, respectively, then ANFIS model is generated using subtractive clustering method has $(n \times m \times p + m \times (n+1))$ parameters whereas an ANFIS model generated using the grid method has $(n \times m \times p + n^m \times (n+1))$ parameters. For example, in order to model a system has 6 inputs and 1 output using 3 membership functions for each input, if the ANFIS model is generated with the subtractive clustering method, it will have 57 parameters, and if it is generated with the grid partition method, it will have 5157 parameters. It is obvious that if the ANFIS model have too many parameters, much training time is needed to identify the parameters. In addition, a model with too many parameters usually faces problem of over-fitting. It is not advisable to use grid partitioning in ANFIS when the input dimension is more than five or six (Nayak *et al.*, 2005), due to curse of dimensionality, which results in an excessive propagation of the number of rules. Therefore the subtractive clustering method was chosen when generating the ANFIS batch (off-line) mode model.

In this study, the number of membership functions applied to each input ANFIS was identified by trial-and-error. The number of membership functions was increased from 3 to 20. It was found that when the number of rules increased from 3 to 10 the performance of the model was more or less the same, and when the number of rules was greater than 10, model performance deteriorated. The number of membership functions was thus selected as 4. This also means that the range of

water levels can potentially be divided into 4 regions: low, medium, high and very high. After training, the membership functions are identified as shown in Fig 5.3 as follows.

Figure 5.3. Membership functions identified for water level at Pakse in ANFIS model.

5.3. Other Models used for Comparison with ANFIS

The present study provides comparisons between ANFIS with Multiple Linear Regression (MLR), Multi-Layer Perceptron Neural Network (MLP), and Unified Run-off Basin Simulation (URBS) models. The principles of these models have been presented in Chapter 2. This section presents how these models have been developed to provide 5-lead-day water level forecasts at Pakse. Then, the 5-lead-day water level forecasts given by these models at Pakse are compared.

5.3.1. Multiple Linear Regression Models

Two Multiple Linear Regression models were utilized for 5-lead-day water level forecasting at Pakse in the present study: (i) first-order and (ii) second-order polynomial models. The first-order polynomial model has regressors in linear form:

$$H_{PK}(t+5) = \beta_0 + \sum_{j=1}^6 \beta_j x_j + \varepsilon \quad (5.3)$$

whereas, the second-order polynomial model has regressors in linear, interaction and square terms:

$$H_{PK}(t+5) = \beta_0 + \sum_{j=1}^6 \beta_j x_j + \sum_{i=1, j=1, i \leq j}^6 \beta_{ij} x_i x_j + \varepsilon \quad (5.4)$$

Here, x_1, x_2, x_3 and x_4, x_5, x_6 are water levels at Pakse and Savannakhet at time steps $t, t-1, t-2$, respectively. These variables are defined in Eqn. 5.2. The linear regression models in the present study are developed based on a regression function in statistics toolbox provided by MatLab (2008).

5.3.2. Multi-Layer Perceptron Neural Network

The present study utilized the Multi-Layer Perceptron Neural Network, available in the Neural Network Toolbox, MatLab (2008). The training method used for the ANN is the Levenberg-Marquardt algorithm. This algorithm approaches second-order training speed without having to compute the Hessian matrix that is required in a standard feedforward ANN. The Levenberg-Marquardt algorithm appears to be the fastest method for training moderate-sized feedforward neural networks (up to several hundred weights) according MatLab (2008). Following the general relationship shown in Eqn. 5.2, the relationship between 5-lead-day water level at Pakse and independent inputs is

$$H_{Pakse}(t+5) = g\{H_{Pakse}(t), H_{Pakse}(t-1), H_{Pakse}(t-2), H_{Sav}(t), H_{Sav}(t-1), H_{Sav}(t-2)\} \quad (5.5)$$

The study developed a 3-layer (i.e. 1 input, 1 hidden, and 1 output layer) MLP Neural Network with the number of nodes in the hidden layer identified by the trail-error method. It was found that training errors decreased continuously when the number hidden nodes increased from 3 to 11. The validation error decreased up to 6 hidden neurons, after which it increased (Fig. 5.4). Therefore, a neural network with 6 hidden nodes was chosen.

Figure 5.4. Training and validation errors corresponding to number of hidden nodes in the MLP neural network.

5.3.3. URBS Model

The Lower Mekong River Basin covers an extensive area and consists of a number of major tributaries whose sub-basins have different hydrological characteristics. Following physically-based modelling approach, modelers at Mekong River Commission have divided the whole river basin from Chiang Sean to Pakse into thirty eight sub-basins, each of which is modeled by URBS sub-models. These models are linked together to provide forecasts of discharges at stations along the main stream. The flow rates at stations are then converted to water levels. The URBS-based flood forecasting system has been in operation since 2009. The comparison between ANFIS and URBS is for 2009 since that is the only year where URBS data are available.

5.4. Output Error Correction Using Autoregressive Models

Assuming that there are five separate ANFIS models denoted as ANFIS_l ($l = 1, 2, \dots, 5$) to provide 1- to 5-lead-day water level forecasts at a gauging station. The output of ANFIS_l at time t denoted by $\hat{H}_t(t+l)$ is the water level forecast made at time t for the actual (measured) water level, $H(t+l)$, at lead time $t+l$. The

difference between $H(t+l)$ and $\hat{H}_t(t+l)$ is the output error of the ANFIS_{*t*} model at time t . This error is denoted by $e_t(t+l)$ and is given by:

$$e_t(t+l) = H(t+l) - \hat{H}_t(t+l) \quad (5.6)$$

At time t , $e_t(t+l)$ is unknown since $H(t+l)$ is unknown. An output updating procedure would attempt to determine $\hat{e}_t(t+l)$ which is an estimate of the actual error, $e_t(t+l)$. The corrected (updated) output $\hat{\hat{H}}_t(t+l)$ is then given by the formula:

$$\hat{\hat{H}}_t(t+l) = \hat{H}_t(t+l) + \hat{e}_t(t+l) \quad (5.7)$$

To update the output of the ANFIS_{*t*} model we need to model the time series $e_t(t+l)$. An Auto-regressive (AR) model (described in Section 3.7.1) is typically used in order to model the output error time-series in real-time hydrological forecasting (Serban and Askew, 1991; WMO, 1992). From Eqn. 3.23, the general form for an AR model of order p , AR(p), modeling the output error time-series $\hat{e}_t(t+l)$ can be expressed as follows:

$$\begin{aligned} \hat{e}_t(t+l) = & \bar{e}_t(t+l) + \phi_1(e_{t-1}(t+l) - \bar{e}_t(t+l)) + \phi_2(e_{t-2}(t+l) - \bar{e}_t(t+l)) + \dots \\ & + \dots + \phi_p(e_{t-p}(t+l) - \bar{e}_t(t+l)) \end{aligned} \quad (5.8)$$

The AR(p) model is calibrated using the output training errors of the ANFIS models, after which it is applied to estimate the testing output errors.

For any ANFIS_{*t*} model, at time t , the most recently known error is $e_{t-l}(t+l) = H(t) - \hat{H}_{t-l}(t+l)$ which is the difference between measured water level at time t or $H(t)$ (current measured value), and the forecast of ANFIS_{*t*} made at time step $t-l$ or $\hat{H}_{t-l}(t+l)$. In order to estimate $e_t(t+l)$ we have to apply the AR model (Eqn. 5.8) l times recursively, i.e., starting with known error $e_{t-l}(t+l)$ to estimate $\hat{e}_{t-l+1}(t+l)$ and so on until $\hat{e}_t(t+l)$ is obtained. Hence, this means that updating the

output of the ANFIS₁ model requires 1-step-ahead error prediction, updating the output of the ANFIS₂ model requires 2-steps-ahead error prediction, and so on. In the present study, the procedure using an AR model in l recursive steps to predict the output error of the ANFIS _{l} model is referred to as a fully recursive output error prediction and is denoted by AR_a. Fig 5.5(a) provides an illustration of the AR_a procedure to predict output error of the ANFIS₅ model where an AR model with order 1, AR(1), model is used. The initial step (Step 0) computes the current known output error of the ANFIS₅ model, $e_{t-5}(t+5) = H(t) - \hat{H}_{t-5}(t+5)$. This error is then applied to the AR model to obtain an estimated error for Step 1. The process is continued until the estimated error $\hat{e}_t(t+5)$ is obtained after 5 recursive steps (at Step 5). Because of the errors accumulated through each recursive step in the AR_a procedure, thus, it can lead to high inaccuracies in $\hat{e}_t(t+5)$.

It is found in the present study (and indeed from literature) that ANFIS model predictions are expected to be accurate for short lead time forecasts. Assuming that the ANFIS _{j} model is able to provide acceptable j -lead-day forecasts, it would thus be reasonable to use the result of ANFIS _{j} model, $\hat{H}_t(t+j)$, to replace the actual unknown value $H(t+j)$ at time t in the estimation of the ANFIS _{l} output error at time $t-l+j$ ($l > j$), $e_{t-l+j}(t+l) = H(t+j) - \hat{H}_{t-l+j}(t+l)$. Hence, for the case where $j = 2$, we can replace $\hat{e}_{t-3}(t+5) = \phi_5 \hat{e}_{t-4}(t+5)$ in the AR_a procedure (Fig 5.5(a)) by $\hat{e}_{t-3}(t+5) = \hat{H}_t(t+2) - \hat{H}_{t-3}(t+5)$ to initiate the AR_b procedure, as shown in Fig 5.5(b). This error prediction procedure is referred to as the partial recursive output error prediction procedure and denoted by AR_b. As a result, the number of recursive steps in AR_a is reduced by 2, as shown in Fig 5.5(b) for the case of $j = 2$. Similar to the AR_a and AR_b procedures, the fully recursive procedure using the RAR model is denoted by RAR_a, and the partially recursive procedure using the RAR model is denoted by RAR_b.

Figure 5.5. The output error prediction procedures for the ANFIS₅ model: (a) AR_a and (b) AR_b procedures

In practice, error prediction may lead to rapid oscillations and/or extreme peaks. Thus, a threshold should be introduced to an output updating procedure to avoid this problem (Lekkas, 2002); the predicted error is applied only if its absolute value is less than a threshold value, ε_{th} . A value of $\varepsilon_{th} = 0.8$ meters was determined by trail-and-error and adopted for this study. The l -step-ahead error prediction procedure

can be either AR_a , AR_b , RAR_a or RAR_b . Therefore, the absolute value of the estimated error at time t is then compared to ε_{th} . If it is less than ε_{th} , the estimated error is used to update the original output, else the original output value is used as the water level forecast.

5.5. Results and Discussions

5.5.1. A Comparison between ANFIS and MLR, MLP Models

The performance of two linear regression models (i.e. first- and second-order polynomial, denoted by REG-1 and REG-2 in Fig.5.6, respectively), ANFIS and ANN models was evaluated based on continuous daily water levels in the flood seasons (June to November) of 11 years (1993 to 2003). This data was divided into training (1993 to 1998), validation (1999-2000), and testing (2001-2003) data sets. The performance indexes were employed including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Coefficient of the Efficiency (CE) or Nash-Sutcliffe coefficient, and Persistence Index (PI).

The comparison is shown in Fig. 5.6. The error indexes show that ANN and ANFIS models perform better than the two linear regression models. Among the four models, the first-order linear regression model is the worst as it takes into account only linear relationship between inputs (Savannakhet - Pakse water levels) and output (the 5-lead-day water level at Pakse). The second-order linear regression model slightly outperforms the first-order linear regression model since it has interaction and square terms of input variables by which some non-linear degree for the relationship between inputs and output are added to the models. It can be observed that the training errors of the linear regression models are significantly higher than that for ANN and ANFIS models. The ANFIS model outperforms the ANN and regression models training, validation and testing. Figure 5.7 shows comparison between 5-lead-day water level forecasts given by ANN and ANFIS models versus observed water levels at Pakse during three flood events (in years 1996, 2000 and 2001, respectively) including the training, validation and testing data sets where water levels were above a medium level (i.e.7.5 m). The results

provided by ANFIS are more reasonable than that given by ANN. In addition, Table 5.2 shows that ANFIS model has lower MAE than ANN model for the given flood events. In addition, ANFIS has smaller absolute errors than ANN for flood peaks prediction.

Figure 5.6. Model performance comparison between linear regression, ANFIS and ANN models for 5-lead-day water level forecasting at Pakse.

Figure 5.7. A comparison between observed water level at Pakse and 5-lead-day forecast given by ANN and ANFIS models

Table 5.2. Comparison between ANN and ANFIS during three flood events in validation and test data set.

Flood events	Duration	Peak (m)	MAE (m)		Error in estimating peak water level (m)	
			ANN	ANFIS	ANN	ANFIS
1 (Training)	13/09/96	13.01	0.57	0.53	-0.68	-0.59
	11/10/96					
2 (Validation)	29/08/00	13.34	0.89	0.77	-1.54	-0.84
	26/09/00					
3 (Testing)	20/07/01	12.70	0.66	0.56	-1.04	-0.73
	01/10/01					

The above results show that ANN and ANFIS models have better forecast ability compared to linear regression models to approximate the highly non-linear relationship between recent water levels, rainfall on sub-basins and the 5-lead-day water level at Pakse. Moreover, ANFIS outperforms ANN on overall evaluation for training, validation and testing data sets and also in peak water level estimation.

5.5.2. Comparison between ANFIS and URBS Models

Five independent ANFIS models, with general input-output relationship as shown in Eqn. 5.2, were developed in MATLAB (2008) to provide water level forecasts at Pakse for lead times from 1 to 5 days. The MAE of the ANFIS results for the entire training (the wet seasons from 1993 to 1998) and testing (the wet seasons from 2001 to 2003 and 2009) datasets is shown in Fig 5.8. Note that the discussion here pertains to the ANFIS model results denoted as “ANFIS”, in Fig 5.8. The discussion on the results for output updating (denoted as ANFIS + Updating in these figures) is presented in the next section. Here, the MAE is used since this is the error measure adopted by the RFMMC for its benchmark of flood forecasts. In Fig 5.8, the MAE of the ANFIS results is significantly less than the MRC’s benchmark for the all lead times.

Figure 5.8 . MAE calculated on the one- to five-lead-day forecasts for entire wet season (with and without updating) for (a) the training data set and (b) the testing data set.

However, since the low and medium water levels (less than 10 meters at Pakse) dominate the dataset, it is necessary to evaluate the MAE of ANFIS results for high water level regimes separately (higher than 10 meters at Pakse), in order to gauge ANFIS model performance for high water levels. The MAE of the ANFIS results for the high water levels is shown in Fig 5.9.

Figure 5.9. MAE calculated on the one- to five-lead-day forecasts for high water level (with and without updating) for (a) the training data set and (b) the testing data set

As shown in Fig 5.9(a) and 5.9(b), the MAE of the ANFIS results for 1- and 2-lead-day forecasts is significantly lesser than the benchmark. This indicates the ANFIS₁ and ANFIS₂ models are able to meet required level of accuracy for high water levels. The ANFIS₃ and ANFIS₄ models are still able to provide reasonable results since their MAE are about the same as the benchmark. However, the ANFIS₅ model has higher MAE than the benchmark for both training and testing datasets. This is in general true for most predictive models; models are able to give good forecasts for short lead times, but their accuracy may be inadequate for longer lead times.

A further analysis on the testing ANFIS results is carried out by a comparison between ANFIS and URBS results for the 2009 wet season (the URBS results are not available for years from 2001 to 2003 since it has been utilized at RFMMC since 2009). This is shown in Fig 5.10, in which the naïve model is used as a baseline for the comparison. The naïve or persistence model refers to a model

that takes the measured water level at the current time step t as a forecast of the water level at the time step $t + l$ where $l = 1, 2, \dots, 5$ time steps ahead.

Figure 5.10. A comparison of performance between Naïve, URBS and ANFIS: (a) COE, (b) MPAE, (c) MAE and (d) RMSE

All the error indexes show that the ANFIS models such as ANFIS₁, ANFIS₂ and ANFIS₃ perform better than the URBS model as providing more accurate 1-, 2- and 3-lead-day forecasts. However, ANFIS₄ and ANFIS₅ models can only obtain a slight improvement in comparison with URBS model for 4- and 5-lead-day forecasts. The ANFIS models developed in the present study rely on the auto-correlation at Pakse and cross-correlation between the water levels at Savannakhet

and Pakse. The better performance of the ANFIS models for 1- to 3-lead-day forecasts is due to the strong correlations for 1 to 3 time lags (Fig 5.1(a) and 5.2(a)); thus the data-driven ANFIS models can approximate well the water levels at Pakse up to 3 lead days using recent information of water levels at Pakse and Savannakhet. However, for 4- and 5-lead-day these correlations decrease significantly; for example, the partial- autocorrelation of water levels at Pakse reduces exponentially and tails off when lag times are longer than 3 days (Fig 5.1(b)). The correlations of the water level time series thus places a practical limit on the forecast lead time achievable with the ANFIS models developed in the present study; due to the decrease in the correlation for $l \geq 3$ days, the ANFIS₄ and ANFIS₅ models do not provide a significant advantage to the URBS model.

5.5.3. ANFIS Results with Updating

Four updating procedures, i.e., AR_a, RAR_a, AR_b, RAR_b (Section 5.4) were developed to enhance the accuracy of water level forecasts at Pakse from 1 to 5 lead days by correcting the original ANFIS_{*l*} results. At a current time step *t*, the fully recursive procedures AR_a or RAR_a requires 1, 2, ... , 5 recursive steps to estimate output error of ANFIS₁, ANFIS₂, ..., ANFIS₅, respectively. As shown in Figs 5.8, 5.9 and 5.10, ANFIS models can provide accurate forecasts up to 2 lead days, hence, it is reasonable to use the output of ANFIS₂ to initiate the error estimation process of the partial recursive procedure (AR_b or RAR_b). By using the ANFIS₂ results, the partial recursive updating procedure requires 1, 2 and 3 recursive steps to estimate output error of ANFIS₃, ANFIS₄ and ANFIS₅, respectively. The performance of each updating procedure was evaluated based on the coefficient of determination, R^2 :

$$R^2 = 1 - \frac{\sum_{i=1}^n (e_i - \hat{e}_i)^2}{\sum_{i=1}^n (e_i - \bar{e})^2} \quad (5.9)$$

where e_i is the actual ANFIS output error, $\hat{e}_i$ is the estimated ANFIS output error, $\bar{e}$ is the mean value of the actual ANFIS output error and n is number of actual ANFIS output error values. A comparison of the R^2 for the various updating procedures is shown in Fig 5.11.

Figure 5.11. A comparison of AR_a, RAR_a, AR_b and RAR_b updating procedures

Figure 5.11 shows that R^2 values of the updating procedures using the RAR model are higher than those using the AR model, i.e., RAR_a performs better than AR_a, RAR_b performs better than AR_b. This means that the adaptive feature of the RAR model provides an advantage over the AR model, where the parameters are fixed after calibration. Figure 5.11 also shows that the R^2 values of AR_b and RAR_b are higher than those of AR_a and RAR_a, respectively. A comparison between the errors estimated by partial recursive procedure $[\hat{e}_{t-3}(t+5) = \hat{H}_t(t+2) - \hat{H}_{t-3}(t+5)]$ and the errors estimated by the fully recursive procedure $[\hat{e}_{t-3}(t+5) = \phi_5 \hat{e}_{t-4}(t+5)]$, where $\hat{e}_{t-4}(t+5) = \phi_5 e_{t-5}(t+5)$, versus the actual error $e_{t-3}(t+5) = H(t+2) - \hat{H}_{t-3}(t+5)$, of the ANFIS₅ model for the 2000 wet season data is shown in Fig 5.12. Figure 5.12 clearly shows the estimated errors obtained by the partial recursive updating procedure are more accurate than the estimated

errors obtained by the fully recursive procedure, where 2 additional recursive steps are required.

Figure 5.12. A comparison between actual and estimated values obtained for the error $e_{t-3}(t+5)$ by (a) RAR_a and (b) RAR_b

Following the results shown in Fig 5.11, the RAR_a updating procedure was selected to estimate the errors of $ANFIS_1$ and $ANFIS_2$, and RAR_b was selected to estimate the errors of $ANFIS_3$, $ANFIS_4$ and $ANFIS_5$. A comparison based on the MAE index for the ANFIS models with/without updating is included in Figs 5.8 and 5.9, and further comparisons based on the COE is shown in Fig 5.13. The outputs of the ANFIS models updated by the RAR_a and RAR_b models are denoted by ANFIS+Updating in the figures. The figures show that $ANFIS_1$ and $ANFIS_2$ results were improved slightly with updating, while improvements to $ANFIS_3$, $ANFIS_4$ and $ANFIS_5$ results are more significant. An explanation for this observation is presented in section on correlation analyses.

Figure 5.13. A comparison of the ANFIS model results without and with updating procedure for 1- to 5-lead-day forecasts for (a) the training data set and (b) the testing data set

The wet seasons in 2000 and 2001 were selected for a further analysis, in order to evaluate the performance of the ANFIS models and the updating procedures developed in the present study for long lead time (5-lead-days) forecasts. The floods for these two years were exceptionally, with the peaks at 13.34 m and 12.63 m, respectively. The plots of measured and computed water level time series of the ANFIS₅ model and of the ANFIS₅+RAR_b model are shown in Fig 5.14. As shown in Fig 5.14, the errors of the ANFIS₅ model are dominated by lag-time errors with the predicted results lagging the observed water levels. In general, it is noted that the lag-time error is more significant for the rising limb than for the falling limb. However, as shown in Fig 5.14, the output updating procedure can reduce the lag time errors of the ANFIS models for the falling limbs considerably, but only slightly for rising limb.

Quantitative analysis on the errors of ANFIS₅ with/without updating for the 2000 and 2001 flood seasons for different flow regimes including the overall flood season, high (> 10 m) water levels, rising and falling limbs is shown in Table 5.3. It is observed that the RAR_b procedure reduces the MAE of ANFIS₅ by 0.09 m and 0.02 m for the overall wet seasons in 2000 and 2001, respectively, and by 0.17 m and 0.05 m for the high water level. In addition, Table 5.3 shows that the RAR_b reduces the MAE for the falling limbs much more than for rising limbs: the MAE

was reduced by 0.20 m for the falling limb and 0.12 m for rising limb for the 2000 flood season, and by 0.23 m for the falling limbs and 0.06 m for rising limbs for the 2001 flood season.

Figure 5.14. Measured and computed water level time series of ANFIS₅ and ANFIS₅+RAR_b for five-lead-day forecasts in (a) year 2000 and (b) year 2001

Table 5.3. MAE (m) of ANFIS₅ output results with/without updating by the RAR_b procedure for the 2000 and 2001 floods.

Regimes	2000 Flood Season		2001 Flood Season	
	ANFIS ₅	ANFIS ₅ +RAR _b	ANFIS ₅	ANFIS ₅ +RAR _b
Overall flood season	0.66	0.57	0.58	0.56
High water levels	0.80	0.63	0.72	0.67
Rising limb	0.93	0.81	0.75	0.69
Falling limb	0.55	0.35	0.57	0.34

5.6. Correlation Analyses

It has been mentioned in the previous section that updating ANFIS₁ and ANFIS₂ did not result in as much improvement as updating the ANFIS₃, ANFIS₄ and ANFIS₅ models. This is because the correlations between the errors $e_{t-1}(t+1)$ and $e_t(t+1)$ for ANFIS₁ and between the errors $e_{t-1}(t+2)$ and $e_t(t+2)$ of ANFIS₂ are generally weaker than the correlations between the errors $e_{t-1}(t+3)$ and $e_t(t+3)$, $e_{t-1}(t+4)$ and $e_t(t+4)$, $e_{t-1}(t+5)$ and $e_t(t+5)$ for ANFIS₃, ANFIS₄ and ANFIS₅, respectively. This is shown qualitatively in Fig 5.15. The scatter plots in Fig 5.15 show the relationship between the values of the error time series at time $t - 1$ and at time t for $e_t(t+1)$, $e_t(t+3)$ and $e_t(t+5)$. The relationship between $e_{t-1}(t+1)$ and $e_t(t+1)$, shown in Fig 5.15(a) is clearly random. The trend becomes more distinct as the forecast lead time increases to 3 and 5 days, indicated by Fig 5.15(b) and 5.15c, respectively. Fig 5.15 thus shows that the correlation between the values of the error time series at time $t - 1$ and at time t increases as lead time increases. As shown in Fig 5.15(c), a clear linear correlation can be observed between $e_{t-1}(t+5)$ and $e_t(t+5)$. The improvements to the correlation between the values of the error time series at time $t - 1$ and at time t observed in Fig 5.15 can be studied quantitatively using the Auto-correlation Function (ACF) and a Partial Autocorrelation Function (PACF). The ACF and PACF of the error time series $e_t(t+1)$, $e_t(t+2)$, ..., $e_t(t+5)$ corresponding to the training output error of ANFIS₁, ANFIS₂, ..., ANFIS₅ are plotted in Fig 5.16.

Figure 5.15. Plots of relationship between (a, b) the errors $e_{t-1}(t+1)$ and $e_t(t+1)$ for ANFIS₁, (c, d) the errors $e_{t-1}(t+3)$ and $e_t(t+3)$ for ANFIS₃ and (e, f) $e_{t-1}(t+5)$ and $e_t(t+5)$ for ANFIS₅

Figure 5.16. Autocorrelation and partial autocorrelation of ANFIS models' training errors at (a) one-lead-day, (b) two-lead-day, (c) three-lead-day, (d) four-lead-day and (e) five-lead-day forecasts

Figure 5.16 shows that the auto-correlation coefficient at lag-time of 1 day increases from 0.23 (Fig 5.16(a)) for the time series $e_t(t+1)$ to 0.84 (Fig 5.16(e)) for the time series $e_t(t+5)$. This is consistent with the observations in Fig 5.15(a) and 5.15(e). In general, Fig. 5.16 shows that the linear correlation between the error values at time $t - 1$ and at time t for the time series output error of the ANFIS_{*l*} models increasing as l increases. As a result, the updating procedures can improve the output results of the ANFIS₃, ANFIS₄ and ANFIS₅ models better than the ANFIS₁ and ANFIS₂ models. Further observations of Fig 5.16 shows that the ACF tails off exponentially to zero while the PACF reach a minimum after a lag-time of 1 day. According to Box and Jenkins (1976), this suggests that the output error time-series can be modeled by either the AR or RAR models of order 1, where the general form of the AR or RAR model use to estimate the time series $e_t(t+l)$ in this study:

$$\hat{e}_t(t+l) = \phi_l e_{t-1}(t+l) \text{ where } l=1, 2, \dots, 5, \quad (5.10)$$

is sufficient.

5.7. Rising Limb

There were eight flood events (water level exceeded 11m) recorded at Pakse during the period 1993 to 2003. The characteristics of the rising limb portion of the water levels for the eight flood events are described in Table 5.4 and includes the length (number of days) of the rising phase, the first and last days of the rising phase, the initial and peak water levels and the rate of increase ($\overline{\Delta H}$), defined as the total increase in water elevation divided by the length of the rising phase. Events 1, 2 and 3 were included as training events, events 4 and 5 were included as validation events, and events 6, 7, and 8 were included as testing events.

CHAPTER 5: THE DATA-DRIVEN APPROACH AS A FLOOD FORECASTING MODEL

Table 5.4 Data on rising limbs for flood events during the period 1993 to 2003.

Events	Number of days	Duration		Water level		Rate of increase (m/day)
		First day	Last day	Initial (m)	Peak (m)	
1	26	12/08/1995	06/09/1995	9.93	12.39	0.09
2	15	12/09/1996	26/09/1996	7.00	13.01	0.40
3	23	12/07/1997	03/08/1997	5.18	12.34	0.31
4	17	03/07/2000	19/07/2000	7.69	11.76	0.24
5	17	29/08/2000	14/09/2000	9.67	13.34	0.22
6	31	20/07/2001	19/08/2001	8.16	12.70	0.15
7	11	03/09/2001	13/09/2001	10.18	12.54	0.21
8	36	18/07/2002	22/08/2002	9.49	12.19	0.08

The MAE of ANFIS model for the rising phase of these eight events are presented in Fig 5.17 as follows.

Figure 5.17. (a) MAE (m), and (b) MP AE (%) of ANFIS₁, ANFIS₃ and ANFIS₅ for 1-, 3- and 5-lead-day water level forecasts of the rising limbs for training (events 1, 2 and 3), validation (events 4 and 5) and testing (events 6, 7 and 8) events.

The 1-lead-day forecasts are very close to the observed water levels and the MAEs for the 1-lead-day forecasts are smaller than MRC's benchmark of 0.25m for all the events. The errors are increased for 3-lead-day forecasts. It can be observed that the

MAEs for the training events increase according to the sequence $0.28\text{m} \rightarrow 0.58\text{m} \rightarrow 0.94\text{m}$ as $\overline{\Delta H}$ increases according to $0.09\text{m/day} \rightarrow 0.31\text{m/day} \rightarrow 0.40\text{m/day}$ for events 1, 3 and 2, respectively. This provides an indication that even for the training events, ANFIS error increases when the water level rise is more rapid. It should be mentioned; however, that the MAE for events 1 and 3 are within MRC's benchmark of 0.75m . The MAEs for event 4 and event 5 (validation events) are 0.93m and 0.67m , respectively. The MAEs for the three testing events are 0.46m , 0.48m and 0.48m for events 7, 8 and 9, respectively, all of which are below the MRC benchmark. This is significant since events 7, 8 and 9 were large flood events where overtopping of the river bank at Pakse occurred. In this respect, the ANFIS₃ model has performed reasonably well for the 3-lead-day forecasts. For 5-lead-days forecasts, the two smallest MAEs ($\text{MAE} = 0.53\text{m}$ and 0.77m) were obtained for the two events (events 1 and 8) with the smallest value of $\overline{\Delta H}$ ($\overline{\Delta H} = 0.09\text{ m/day}$ and 0.08 m/day). The MAEs for the other events were considerably higher than the benchmark of 0.75 m .

The dependence of ANFIS forecast error as a function of $\overline{\Delta H}$ is shown in Fig 5.18. Figure 5.18 shows that, in addition to the lead time, the MAE of ANFIS model forecasts depends on $\overline{\Delta H}$, although this observation is less distinct for 5 days forecasts due to the scatter in the data. Thus, the requirement of the model to provide accurate long term forecasts under conditions of rapid water level rise is critical. In this respect, compared with MRC's benchmark of 0.75m for 3 and 5 days forecasts, the ANFIS model is able to provide acceptable forecasts if the $\overline{\Delta H} \leq 0.3\text{ m/day}$ for 3 days forecasts, and for 5 days forecasts, except for the case where $\overline{\Delta H}$ is small ($\overline{\Delta H} \leq 0.1\text{m/day}$), ANFIS₅ model performance is fair.

Figure 5.18. ANFIS model forecast accuracy as a function of ΔH .

Figure 5.19 shows the measured and 1, 3 and 5 days forecasted water level for events 4, 5, 6, 7 and 8. It can be observed that the progressive increase in ANFIS forecast errors is due to phase shift or timing errors. The phase shift errors increased when the lead-time of forecasts increased. The reason of phase shift errors is mainly due to the use of previous measured water level as inputs for ANFIS model.

Figure 5.19. Comparison between the 1-, 3- and 5-lead-day water level forecasts with observed water levels of the rising limbs for validation (events 4 and 5) and (events 6, 7 and 8) testing events.

It was found that previous measured values of water levels are usually highly correlated with the present values. This is commonly referred to as the autocorrelation effects and are related to memory effects in river flow measurements. For example, Fig 5.1 shows significant autocorrelation of the water level time series Pakse (The significant autocorrelation is also presented in Fig 6.1 for Thakhek and Kratie stations). In addition, previous measured water level values can be considered indirect indicators of the hydrological state of a catchment such as recently high water level indicates catchment has just received high rainfall and soil moisture could be high. For this reason, the measured water levels or discharge are often used as model inputs for ANN (Hsu et al., 1995; Minns and Hall, 1996; Campolo et al., 1999), NFIS (Nayak et al., 2005; Chen et al., 2006) and SVM (Yu et al., 2006) models. The present study also uses previous water levels as input information for ANFIS model to forecast water level in next few days. However, as pointed out by de Vos and Rientjes (2005), the autoregressive component can become too dominant and it leads to a forecast that is too similar to the last known value, resulting in timing errors or phase lag in a model's forecasts. Their study suggested that if moving average of previous discharge/water level was used instead of direct previous discharge/water level then the timing error could be decreased. This issue has also been discussed by Abrahart et al. (2007), but it is generally accepted that timing errors remain a problem and no method exists that can completely eliminate this problem.

5.8. Conclusions

The following can be concluded from this study:

1. The ANFIS models provided good performance for forecasting one-, two- and three-lead-day water levels at Pakse of the Lower Mekong River as they satisfy the benchmark when the models were evaluated on the entire data set as well as at high (> 10 m) river stages.
2. The Comparison between ANFIS and MLR, MLP model's results for years 1993 to 2003 show that ANFIS is superior compared to these models.

3. Comparing between ANFIS and URBS models' results for the year 2009, the ANFIS model produced better results compared with the URBS model for one- to three-lead-day forecasts. However, ANFIS archived only slight improvement in comparison with URBS model for four- and five-lead-day forecasts.

4. Considerable improvements to three-, four- and five-lead-day forecasts were obtained when ANFIS model results were updated by the output updating procedures. Among the developed updating procedures, the partial recursive output error prediction procedure, RAR_b , which uses the RAR model to approximate errors and uses the ANFIS2 model results in initiate the error estimation process, gave the best results.

5. The study highlights that the challenge for water level forecasting at Pakse for long lead times remains. At long lead times, model errors are dominated by lag time errors with the predicted results lagging the observed water levels. Although the partial recursive output error predicting procedure provided the best results, this method was able to reduce the lag time errors of the ANFIS models for the falling limbs considerably, but only slightly for rising limb.

CHAPTER 6

WATER LEVEL FORECASTING IN LARGE RIVER SYSTEMS: INPUT SELECTION FOR DATA-DRIVEN MODELS

6.1. Introduction

The input requirements for physically-based models are known *a priori*. Often this would include the physical properties of the catchment such as the watershed area, land use, type of surface cover, slope, soil properties, in addition to the measured rainfall and discharge. Data driven models however are less constrained in terms of data requirements. The minimum requirement is the time series data of water levels (Liong et al., 1996); however, this can be augmented by other data such as rainfall (Chen et al., 2006) and temperature (Dawson and Wilby, 1998) if these data are available. For a large river, the input requirement for flow forecasting at different stations will be different due to differences in flow behavior. Thus, models inputs in relation to the considerations mentioned above need to be evaluated. In addition, flood forecasts are required at a number of gauging stations along the main stream. However, the flow at these stations to be modeled can differ in terms of the flow contributions from upstream stations and contributions from rainfall and lateral flows from sub-basins immediately upstream of the station of interest. These factors would invariable give rise to differences in the inputs to be adopted. As shown in the review presented above, studies carried out on flood forecasting with data driven models are usually focused on model performance and investigations into the link between the flow behavior and input requirements are rare. In addition, most forecasting studies are carried out for short term forecasts (one or two time steps ahead). In order to assess the utility of data driven models, however; moreover studies focusing on the accuracy of data driven models in long term forecasting need to be carried out.

Some studies have been carried out developing data-driven models for large river systems. Liong et al. (1996) developed ANN model to forecast up to 7 lead

days the water levels in Dhaka, Bangladesh which is situated close to the confluence of three large rivers, the Ganges (length = 2,525 km), the Brahmaputra (length = 2,900 km), and the Meghna (length = 946 km). The authors, used water levels at Dhaka and at three upstream gauging stations, with distances to Dhaka ranging from 160 to 355 km, as inputs to their ANN models. They reported that the ANN model demonstrated a high degree of accuracy, even for up to 7 lead days. Antar et. al. (2006) and Shamseldin (2010) developed ANN models to forecast the 1 day ahead discharge for the Blue Nile River. Antar et al (2006) divided the basin at Diem (drainage area = 142,000 km²) into 7 sub-basins and used the areal average rainfalls and a number of previous observed discharges at Diem as inputs. The authors compared the ANN model results with a physically-based distributed model and found that the ANN models were unable to explain only 3% of the variability of observed data, compared to 7% for the physically based model. Shamseldin (2010) divided the El Diem basin into sub-basins and used a rainfall index which is a weighted sum of the recent and current rainfalls, the seasonal expectation rainfall index, and the seasonal expectation of the discharge as model inputs. The author reported that model performance reduced when forecast lead-times increased from 1 to 6 days. However, applying an output updating technique improved model accuracy. Dawson et al. (2002) evaluated the multilayer perceptron (MLP) and radial basis function network (RBF) to forecast the flow at the Three Gorges Dam in the Yangtze River, China. The authors used the discharges at the dam, two upstream stations and precipitation as inputs. Both the MLP and RBF networks provided accurate 48 hours ahead forecasts. Solomatine and Xue (2004) used M5 Model Trees and ANN to forecast 1-lead-day discharge at the Xixian station, situated on the Huai River (drainage area = 10,190 km²). The daily areal average rainfall, moving average of areal average daily rainfall, the discharges at Xixian and an upstream station with different corresponding time lags were used as input variables. The results from the M5 model trees and ANN models were good, and significantly, the authors suggested that such techniques can complement physically-based models.

In each of these studies, there was only one forecast station on the main stream and inputs for the model were identified based on prior knowledge of the river system's characters and they were appropriate for the particular forecast station. In practice, flood forecasting in a large river usually requires providing water level forecasts at a number of stations along the river at various time steps (i.e. from one to several days). Therefore, it is beneficial to investigate how various potential input variables influence the performance of a data driven model due to hydrological condition of the forecast station and due to lead time of forecast as well.

Chapter 5 has shown the applicability of ANFIS model for water level forecasting at Pakse station on the main stream of the large river system Mekong. This model's inputs were water levels at Pakse itself and at its upstream station Savanakheth. It was found that 1- and 3-lead-day water level forecasts for year 2009 data provided by ANFIS were more accurate than that were provided by URBS model. The 5-lead-day water level forecasts given by ANFIS have considerable lag time errors, however, their accuracy was comparable to such accuracy the URBS obtained. A question arises here, i.e. if the use of other input variables such as water levels at tributaries, rainfall on sub-basins can improve model performance. If so, what is the degree to which these kinds of inputs can influence ANFIS model performance? In order to answer these questions, the input factors which influence model performance were studied by an investigation of the forecasts of water levels at Thakhek, Pakse and Kratie, three stations along the main stream of the Lower Mekong River. These locations are hydrologically distinguishable from each other in terms of the flow contributions from major tributaries upstream of these stations and the flood travel times from the upstream stations to these locations. Forecasts of 1-, 3- and 5-lead-days were considered in order to investigate how input factors affect to model performance due to lead time of forecast. Although input variable selection issues were investigated for ANFIS model, the findings would be equally applicable to other types of data driven models.

In order to identify appropriate input variables for ANFIS model at each station, the present study has developed four groups of ANFIS models with four

different input variable sets (i.e., only water level at the station of interest, upstream water levels, water level at main tributaries, and rainfalls). Model comparisons are carried out for each of Thakhek, Pakse and Kratie stations and conclusion about the role of the inputs for Thakhek, Pakse and Kratie stations are provided. The ANFIS models presented in this chapter work in the batch (off-line) mode. In this mode, the model is trained based on a given training dataset and its parameters are fixed during the model implementation phase.

6.2. Input Selection of ANFIS Models

The auto-correlation of the water level time series at Thakhek, Pakse, and Kratie is shown in Fig. 6.1(a). An analysis of Fig 6.1(a) suggests that the water levels at each station can be modeled by an autoregressive process of order p , or $AR(p)$. To identify the order p , the partial autocorrelation is used since partial autocorrelation coefficients of an $AR(p)$ process approach zero for lags larger than p (Box and Jenkins, 1970). Fig 6.1(b) shows the partial autocorrelation coefficients of the time series of water levels at the three stations approaching 0 after 3 time lags. Thus, model M1, which uses the water levels at the station of interest only can be denoted as:

$$M1: H(t+L) = f\{H(t), H(t-1), H(t-2)\} \quad (6.1)$$

where $L = 1, 3, 5$ days is the forecast lead time.

Figure 6.1. Correlogram of the water level time series at Thakhek, Pakse, and Kratie: (a) autocorrelation (ACF), and (b) partial autocorrelation (PACF) of water level time series.

Figure 6.2. Cross correlation between water levels at (a) Thakhek (b) Pakse (c) Kratie and water levels at upstream stations.

The cross-correlation of the water level time series at Thakhek, Pakse, and Kratie with the water levels at upstream stations is shown in Fig 6.2(a), 6.2(b) and 6.2(c), respectively. The plots indicate that water levels at Thakhek, Pakse, and Kratie are most correlated with water levels at the stations immediately upstream, i.e. Nong Khai, Thakhek and Pakse respectively. However, their correlations with water levels at further upstream stations are also high (greater than 0.95 for lags of up to 10 days). Therefore, we applied a stepwise approach to identify the potential upstream stations where water level inputs should be included and the lag times of these inputs. In this approach, starting from the immediate upstream station, water level inputs, up to 3 lag times were introduced one by one. If a newly added input improved the model performance then it was considered as significant and was included to the set of inputs, and the next input was then considered, else it was rejected and the procedure was terminated.

The general form of model M2 which includes water levels at upstream stations is denoted by:

$$\begin{aligned} \text{M2: } H(t+L) = f\{ & H(t), H(t-1), H(t-2), H_{\text{up}}^1(t), H_{\text{up}}^1(t-1), H_{\text{up}}^1(t-2), \dots \\ & H_{\text{up}}^k(t), H_{\text{up}}^k(t-1), H_{\text{up}}^k(t-2)\} \end{aligned} \quad (6.2)$$

where H_{up}^k is the water level measured at the k^{th} upstream station.

In addition to the water level measured along the main river, the contribution from the tributaries should also be considered. The model which includes the water levels in the tributaries is denoted as M3:

$$\text{M3: } H(t+L) = f\{ H(t), H(t-1), H(t-2), H_{\text{up}}^1(t), H_{\text{up}}^1(t-1), H_{\text{up}}^1(t-2), H_{\text{up}}^k(t), H_{\text{up}}^k(t-1), H_{\text{up}}^k(t-2), H_{\text{T}}^1(t), H_{\text{T}}^1(t-1), H_{\text{T}}^1(t-2), \dots, H_{\text{T}}^n(t), H_{\text{T}}^n(t-1), H_{\text{T}}^n(t-2)\} \quad (6.3)$$

where $H_{\text{T}}^1 \dots H_{\text{T}}^n$ is the water level in a tributary and $n = 4, 3$ and 2 for Thakhek, Pakse and Kratie, respectively, is the number of tributaries considered. The average rainfall at the sub-basins was next considered. The Thiessen polygon method was utilized to provide areal average rainfall, after which the average rainfall was smoothed by the moving average method. Smoothed rainfall inputs are desirable because rainfall inputs are usually noisy which could have an undesirable artifact on the modeling results. The cross-correlation between the water levels at Thakhek and average rainfall of Sub-basin 1 (in Fig 4.1(c)), between the water levels at Pakse and average rainfall of Sub-basins 2 and 3 (in Fig 4.1 (d) and (e)), and between the water levels at Kratie and the average rainfall for Sub-basins 4 (in Fig 4.1(f)) are shown in Fig 6.3. In the present study, we used three smoothed time series denoted by P_{w3} , P_{w5} and P_{w7} , obtained by moving averages with window size of 3, 5, and 7, respectively. This means, at each of time step, average rainfalls for durations of last 3, 5 and 7 days were taken into account. Fig 6.3 shows that the correlation between rainfall and water level for Thakhek is much higher than Pakse and Kratie implying that rainfall could be a more important factor to consider for Thakhek than Kratie and Pakse. In addition, the maximum correlation generally occurs between 6 to 9 lags indicating that the flows resulting from the rainfalls in the sub-basins take about 6 to 9 days to reach the stations considered. ANFIS models whose inputs include water levels and the areal average rainfall calculated based on 3, 5 and 7 moving average windows are denoted as M4 models:

$$\text{M4: } H(t+L) = f\{ H(t), H(t-1), H(t-2), H_{\text{up}}^1(t), H_{\text{up}}^1(t-1), H_{\text{up}}^1(t-2), H_{\text{up}}^k(t), H_{\text{up}}^k(t-1), H_{\text{up}}^k(t-2), P_{w3}(t+L-5), P_{w5}(t+L-5), P_{w7}(t+L-5)\} \quad (6.4)$$

Figure 6.3. Cross correlation between average areal rainfall in sub-basin 1 and water level at Thakhek; sub-basin 2 and water level at Pakse; sub-basin 3 and water level at Pakse; sub-basin 4 and water level at Kratie. (a) 3 point moving average, (b) 5 point moving average and (c) 7 point moving average.

A summary of the models adopted in this study is shown in Table 6.1.

6.3. Model Structure Selection and Implementation

The ANFIS Editor GUI, which was used in the application presented in Chapter 5, is also used for the application presented in this Chapter. Model structure selection is an important issue in data-driven modelling. For example, in ANN applications, the number of hidden nodes has an influence on the non-linear approximation ability of an ANN model. The more nodes in the network, the more complex the ANN model is. If an ANN model is too complex, while available data is inadequate then it may lead to problems of overfitting. The complexity of an ANFIS model depends on number of rules, number of membership functions applied to each input and the partitioning method adopted. A grid partitioned ANFIS model has a number of rules that increases exponentially according to the number of n -permutations of a set of m elements given by:

$$P(m,n) = m^n \quad (6.5)$$

where $P(m,n)$ is the number of rules, m is the number of membership functions and n is the number of inputs. The number of parameters in an ANFIS model is:

$$Para = n \times m \times p + r \times q \quad (6.6)$$

where $Para$ is the total number of parameters in the ANFIS model, p is the number of parameters for the membership function (e.g., 2 for Gaussian function and 3 for Bell Shape functions), r is number of rules which equals m for subtractive partitioning and equals m^n for grid partitioning method, and q is the number of parameters in the consequence part of a rule. For instance, if an ANFIS model which has 12 inputs, with 3 membership functions applied to each input is based on grid partition method then the number of parameters is 6,908,841 whereas it is 147 parameters if the ANFIS is based on subtractive partition method. Therefore, the grid partitioning can result in overfitting and require longer time for training. Hence, ANFIS models presented in Chapters 5 and 6 are based on the subtractive partition method. The number of membership functions applied to each input for ANFIS models was identified by the trial-and-error method and the number of membership functions chosen was 4. This also means that the range of water levels can potentially be divided into 4 regions: low, medium, high and very high. The inputs used and structure of ANFIS models adopted in the following Table 6.1.

CHAPTER 6: WATER LEVEL FORECASTING IN LARGE RIVER SYSTEMS - INPUT SELECTION FOR DATA-DRIVEN MODELS

Table 6.1. Inputs used and structure of ANFIS models adopted.

Models	Inputs	Outputs	MFs	Description of inputs
M1	3	1	4	H at $t, t - 1, t - 2$
M2				
Thakhek	6	1	4	$H_{\text{Nong Khai}}, H_{\text{Thakhek}}$, at $t, t - 1, t - 2$
Pakse	6	1	4	$H_{\text{Pakse}}, H_{\text{Thakhek}}$, at $t, t - 1, t - 2$
Kratie	6	1	4	$H_{\text{Kratie}}, H_{\text{Pakse}}$, at $t, t - 1, t - 2$
M3				
Thakhek	18	1	4	$H_{\text{Thakhek}}, H_{\text{Nong Khai}},$ and $H_{\text{BaNaLuang}}, H_{\text{MuongMay}}, H_{\text{MuongBori}}, H_{\text{BanSigno}}$ at $t, t - 1, t - 2$
Pakse	15	1	4	$H_{\text{Pakse}}, H_{\text{Thakhek}}$, and $H_{\text{KeangSaphu}}, H_{\text{SeBangFai}}, H_{\text{BanKengDone}}$ at $t, t - 1, t - 2$
Kratie	12	1	4	$H_{\text{Kratie}}, H_{\text{Pakse}}$, and $H_{\text{Chantangoy}}, H_{\text{BanKamPhun}}$ at $t, t - 1, t - 2$
M4				
Thakhek	9	1	4	$H_{\text{Thakhek}}, H_{\text{Nong Khai}}$, at $t, t - 1, t - 2$, and $P_{w3}(t + L - 5), P_{w5}(t + L - 5), P_{w7}(t + L - 5)$ of rainfall on right side sub-basins.
Pakse	12	1	4	$H_{\text{Pakse}}, H_{\text{Thakhek}}$ at $t, t - 1, t - 2$, and $P_{w3}(t + L - 5), P_{w5}(t + L - 5), P_{w7}(t + L - 5)$ of rainfall on right and left side sub-basins separately.
Kratie	9	1	4	H, H_{Pakse} , at $t, t - 1, t - 2$, and $P_{w3}(t + L - 5), P_{w5}(t + L - 5), P_{w7}(t + L - 5)$ of rainfall on right side sub-basins

6.4. Results and Discussion

6.4.1. Forecast Results for Model M2

We applied a stepwise model building approach to identify the significant inputs for M2 models which use water levels in the main stream as inputs. The PI obtained from this analysis is shown in Fig. 6.4(a), 6.4(b) and 6.4(c) for Thakhek, Pakse and Kratie, respectively. In each of these figures, the value of PI shown at the extreme left-most part of the figure are the results for the model M1, i.e. where only the water level at the station of interest is used as input. It was shown in Nguyen and Chua (2011b) that when only the local information is used, forecast results are severely affected by auto-regressive effects, thus, the forecast accuracy of model M1 is poor. We found that the flows at the stations immediately upstream contribute a major portion (greater than two-thirds of the total flow, see Fig 4.2 in Chapter 4) to the flows at the stations of interest. Water level at the immediate upstream stations is well correlated with the water level at the stations of interest (shown in Fig 6.2). It therefore is common to include information from an immediate upstream station. Figure 6.4 shows that model performance for all stations increases significantly when water levels at the station immediately upstream is included (2 stations in total). However, when the water level from the next upstream station (3 stations in total) is included, the results become mixed. In Fig 6.4(a), the PI for 1 lead day forecast is not increased when water levels at Luang Prabang are included but the PI for 3 and 5 lead day forecasts are increased. In Figs. 6.4(b) and 6.4(c), no appreciable increase in PI can be observed. This is due to the differences in hydrological characteristics of forecast stations.

In general, the river flows at Thakhek, Pakse and Kratie are mainly contributed by the river flow at their immediate upstream stations. However, as shown in Fig 4.2, the contributions are different, i.e. 55% of flow at Thakhek, 84% of flow at Pakse and 63% of flow at Kratie are contributed by the river flow at NongKhai, Thakhek and Pakse, respectively. In case of water level forecasting for Pakse, the flow at Thakhek contributes to 84% the flow at Pakse, the information about water levels at Thakhek is thus the major input information for the ANFIS model to forecast water levels at Pakse. The flow at NongKhai is 45% less than the

flow at Thakhek therefore information about water levels at NongKhai is significantly less informative than information that can provide by water levels at Thakhek in order to forecast water levels at Pakse. Therefore, additional information about water levels at NongKhai did not improve model performance at Pakse (as shown in Fig. 6.4(b)).

It can be concluded that additional information about water levels beyond the station located immediately upstream of the forecast point may not improve model performance if the flow of the station located immediately upstream of the forecast point represent a major portion of the flow at the forecast station. In addition, it was found that when the water level from 3 upstream stations are included (4 stations in total), the PI for all predictions shown in Fig 6.4 decreased significantly. Therefore, it is not necessarily to include water levels at additional stations upstream.

Figure 6.4. Model performance for Model M2. (a) Thakhek, (b) Pakse, and (c) Kratie.

6.4.2. Forecast Results for Model M3

For the reach Luang Prabang - Thakhek, we used water levels of four major tributaries at their gauging stations located at Ba Na Luang (6%), Muong May (2%), Muong Bori (2%) and Ban Signo (3%) as additional inputs for the Thakhek M3 model. For the reach Thakhek - Pakse, water levels at three major tributaries located at Keang Sa Phu (9%), Se Bang Fai (3%) and Ban Keng Done (5%) were

included for the Pakse M3 model. Lastly, for the reach Pakse - Kratie, water levels of two major tributaries located at Chantangoy (9%) and BanKamPhun (8%) were used as additional inputs for the Kratie M3 model.

Figure 6.5. Comparison between M3 and M2 models. (a) Thakhek, (b) Pakse, and (c) Kratie based on RMSE, CE and PI.

Fig 6.5 shows that M3 models have higher RMSE values than M2 models for all 1-, 3-, and 5-lead-day forecasting of water levels at Thakhek and Pakse gauging stations, while slightly lower RMSE values for Kratie gauging station. Consistent with the RMSE, CE and PI for M3 models for Thakhek and Pakse (Fig

6.5(a) and 6.5(b)) are also smaller than that of M2 models, and slightly higher for Kratie (Fig. 6.5(c)). Thus, the addition of water levels measured at the tributaries did not improve model performance for Thakhek and Pakse and only slightly improved results for Kratie. There are several possible reasons to explain the ineffectiveness of including the lateral flows to improve model forecasts. Firstly, the flow contributions represented by the laterals may not be significant, since there is a large volume of flow contributed directly from the catchments into the main stream that are not measured. Secondly, the travel times for flows from the tributaries to the stations of interest is short (we found that the time to peak of water level at the tributary outlets to be the same time as that at the respective stations during flood events). Hence, the inclusion of the lateral flows may not provide additional information to significantly improve long lead time forecasts. Thirdly, we found that while the addition of water levels at the outlets of tributaries reduced training errors, the testing error increased. One possible explanation for this could be that when the flow from the tributaries are included in the model, the time series of water levels may not be entirely independent, since the flow may be the result of the same rainfall event. Therefore, these additional inputs may have the opposite effect of providing extraneous information resulting in a reduction in model performance. However, it should be noted that Kratie, where modest improvements were obtained, has the highest lateral flow contributions of the three stations considered.

6.4.3. Forecast Results for Model M4

M4 models take rainfall as additional inputs beside water levels at the immediate upstream station. The comparison between M4 and M2 model results is shown in Fig 6.6.

For Thakhek, Fig 6.6(a) shows that M4 models did not provide better results compared to M2 models for 1- and 3-lead-day forecasts. However, the M4 model clearly superior to the M2 model for 5 lead-day forecast. The significance of rainfall inputs for Thakhek can be understood by the hydrological characteristics of the Thakhek sub-watershed which is characterized by steep gradients and high average

rainfall (annual rainfall is approximately 3,000mm, see Fig 5.1 in Chapter 5). The cross-correlation between areal average rainfall in sub-basin 1 and water level at Thakhek is thus high as shown in Fig 6.3. Significantly, maximum correlation is achieved at a large time lag, i.e. 5 to 10 days, as shown in Fig 6.3, implying that rainfall can significantly improve long term forecasts (in contrast with M2 and M3 models). In comparison, for Pakse (Fig 6.6(b)) and Kratie (Fig. 6.6(c)), including rainfall information in the model did not help to significantly improve model performance. We found the cross-correlation between rainfall in sub-basin 4 and water level at Kratie, and between sub-basins 2 and 3 with Pakse, is less than that between rainfall in sub-basin 1 and water level at Thakhek (Fig 6.3). The reason is that the annual rainfall is less (1500 – 2500 mm, 2000 – 2500 mm and 1000 – 2000 mm in Sub-basins 4, 2 and 3, respectively) (see Fig 4.1 (b)) and these Sub-basins have generally flatter gradients (see Fig 4.1(a)). This clearer distinction between rainfall and water levels for Thakhek results in better model performance since data driven models work on the basis of modeling model input-output relationships.

Figure 6.6. Comparison between M4 and M2 models. (a) Thakhek, (b) Pakse, and (c) Kratie.

6.5. Conclusions

The study has demonstrated the applicability of ANFIS for flood forecasting of the Lower Mekong which is the lower part of the large river Mekong. Three stations selected in the study, i.e., Thakhek, Pakse and Kratie, are key stations along the main stream since these locations present the significant increases of the main stream by major tributaries of the river system. The study has showed how ANFIS model perform due to the lead time, the location of forecasts and the inputs applied to the models as well. In general, forecast errors is expected to increase when the

variation of a forecasted variable increases. It was found in the present study that the magnitude of changes in water levels increase significantly when the lead time increase from 1 to 5 days and from Thakhek to Pakse and Kratie. Therefore, the ANFIS models' errors increase when the lead time of forecast increases from 1 to 5 days. The lower errors happened when forecasting water levels at Thakhek but higher when forecasting water levels at Pakse and even more at Kratie.

The mean daily flow at a station on the main stream of the lower Mekong during flood seasons is very large. The flow at these stations is mainly contributed by the flow from its immediate upstream station and to a lesser extent by the contribution from lateral flows along the reach between the upstream and downstream stations. Correlation analysis shows that in a number of large flood events, the peak of flow on the main stream occurs more than 4 or 5 days after the storm event. Therefore, 1- to 3-lead-day forecasts are dominated mainly by the travel time of the flood wave between the gauging stations. Hence, addition of rainfall information of sub-basins might not be necessary for 1 to 3 lead-day water level forecasts. Rainfall information however can significantly improve ANFIS model performance for up to 5-lead-day forecasts at locations where rainfall produces significant lateral inflows. This is true for Thakhek where the total flow is contributed almost equally from its immediate upstream station (NongKhai) and sub-basin inflows (Fig. 4.1(c)). For this case, our analysis shows that rainfall inputs can significantly improve ANFIS model performance for 4- and 5-lead-day forecasts as shown in Fig. 6.6(a).

The ANFIS models developed in the study have archived good performance at Thakhek and Pakse since their MAEs is less than the MRC benchmarks for all 1-, 3 and 5-lead-day forecast. At Kratie, the ANFIS model met the MRC benchmark for 3-lead-day forecast. However, it is notice that the benchmark at Kratie is less than for Thakhek and Pakse, but the magnitude of water level changes at this station is significant larger than that at Thakhek and Pakse. When input variables are identified properly and sufficient data is available, ANFIS model for flood forecasting can be setup quickly. Its parameters are optimized automatically and very fast. In this approach, one does not need to identify parameter that related to

characters of catchment and river system which is usually time consuming in physically-based approach.

In the light of conclusion it is recommend that: (i) ANFIS is a promising model to forecast water levels during flood season on a large river such as Mekong, (ii) identifying appropriate inputs for model is the most important task in the model identification and an input can have different role due to location and lead time of forecasts, (iii) when evaluate a flood forecasting model for a large river, we need to select a number of challenge stations in the sense they present the contribution of major tributaries of the river system and the variation in water levels at these station are considerable, thus, evaluation will be more reliable.

CHAPTER 7

APPLICATION OF ON-LINE MODELS FOR FLOOD FORECASTING

7.1. Introduction

In the literature, data-driven models such as ANN and ANFIS are often applied in the batch or off-line mode. In this approach, the training dataset consists of all available input-output training data samples which represent the input-output domain. After training, the model is expected to have the generalization ability of the input-output relationship, and is applied for a testing dataset. However, in practice the training data usually can only represent a limited part of the input-output domain, for example, in flood modelling the training dataset may consist of data for a limited number of flood events. Moreover, individual flood events may have their unique characteristics due to differences in catchment hydrology and variations in rainfall patterns. Thus, the generalization of the trained model in batch mode may not be sufficient and an on-line learning algorithm may be better suited to model future flood events. The TSK fuzzy inference system can be considered as a local linear model since each of its rules define a local input domain where a linear function is applied for the input-output relationship (Vernieuwe, 2005). Hence, on-line learning for a TSK based neuron-fuzzy system is also referred to as “on-line or local learning” in the literature (Nelles, 1998). This learning paradigm is vital to the on-line parameter identification for systems with changing characteristics (Jang, 1993).

In on-line learning, model parameters are updated when new data are presented. Therefore, at the beginning, if available data is scarce, the ANFIS model has to be trained step-by-step by the on-line learning algorithm. In this algorithm, when a new input-output pair is presented, the error between the computed model output and the actual (measured) values is first determined. The error is then used as feed-back information to adjust model parameters (consequence and premise

parameters). The difference between on-line and off-line learning algorithm of ANFIS is that the on-line learning procedure takes into account current model's errors to update the model's parameters when it is in operation, whereas the off-line learning takes into account only the model's errors for a given set of training data samples. Hence, on-line learning tries to adjust model parameters at each time-step when new data becomes available, whereas off-line learning tries to optimize the model's parameters by decreasing the total model error on the whole given training data set and these parameters are then fixed after training. On-line models are especially useful in situations where there is initially no data and the model has to be trained from the beginning. In this case, the batch learning paradigm is not tenable and the on-line training paradigm is a viable alternative. In flood modelling problems where only a limited number of past flood events are available for training a model, it is reasonable to apply a combined training paradigm. In the combined paradigm, off-line training is first used to develop a preliminary model based on a limited dataset, after which model training is extended to the on-line mode, where model adaptation takes place in real-time. This approach is advantageous since model parameters identified in the off-line training phase can be used as good initial estimates for the parameters to be adapted during the on-line phase. Significantly, this learning strategy places less demands on the amount of training data required, compared to models trained in the off-line mode.

This chapter presents a case study of water level forecasting at Thakhek, Pakse and Kratie where the on-line training paradigm was applied to the ANFIS model. The ANFIS model was firstly trained by an off-line learning algorithm. The identified model parameters in the off-line training phase were then used as the initial parameters for the on-line training phase. The on-line and off-line ANFIS models in the present study were implemented using the open source code for ANFIS, in C programming language, which was published by Jang (1998) and is available in <http://www.cs.nthu.edu.tw/~jang/publication.htm>.

7.2. On-line ANFIS Model

The combined training procedure is shown in Fig 7.1. In this method, the ANFIS model is first trained by the off-line learning algorithm. In the initial step of the off-line training phase (Fig. 7.1), the premise parameters of the ANFIS are identified by a grid partition method by which the input variables are divided into equal sub-domains (the number of sub-domain for an input variable equals to the number of membership functions applied on this variable) and the center of membership functions are initiated at the centers of those sub-domains. In the next step, the premise and consequence parameters are optimized based on the hybrid learning scheme with the batch (off-line) mode presented in Section 3.3. After this phase, the on-line learning algorithm presented in Section 3.4 is applied.

Figure 7.1 The hybrid learning paradigm for on-line learning applications

7.3 Water Level Forecasting Using ANFIS with On-line Training

This section investigates the implementation of on-line ANFIS. It was found in Chapter 6 that the important inputs for an ANFIS model to forecast water level at a station along the Lower Mekong are current water levels at the station of interest and at its immediate upstream station and the average rainfall in the sub-basin. The off-line ANFIS models which uses these inputs is model M4 (Section 6.2). A drawback of the grid partitioning method applied for on-line ANFIS developed by Jang (1993) is that the number of rules increase exponentially according to the number of n -permutations of a set of m elements as mentioned in Section 6.3 of Chapter 6. In order to reduce the number of parameters in the on-line ANFIS model, the number of inputs were reduced to three, consisting of two variables, $H(t)$ and $H_{up}(t)$ which are water levels at the station of interest and its immediate upstream station, and the third variable was the 3-days average rainfall at 5-lag-days, $P_{w3}(t+L-5)$. The output variable is the water level at the station of interest at L days ahead. The reduction in the number of inputs should not result in any loss in generality by the models since; the same set of inputs will be used in the comparison between on-line and off-line models, as elaborated in Section 7.4. According to Eqn. 6.5 the number of parameters will be reduced to 135 parameters. Thus, ANFIS models to predict the 1-, 3- and 5-lead-day water levels at Thakhek, Pakse and Kratie with three inputs and one output as mentioned above will have the general input-output relationship as follows:

$$H(t+L) = f\{P_{w3}(t+L-5), H_{up}(t), H(t)\} \quad (7.3)$$

where $L = 1, 3$ and 5 lead days.

Two types of ANFIS models were developed. The input domain of the first model, denoted by M4b₁, is partitioned by the grid partitioning method and is trained by the off-line learning method. The second model, denoted by M4b₂ also uses grid partition method, but is trained on-line, implementing the methodology developed in Section 7.2. Three membership functions were applied for each input, dividing the input domain into 3 regimes: low, medium and high. The structure of the two ANFIS models is summarized in Table 7.1.

CHAPTER 7: APPLICATION OF ON-LINE MODELS FOR FLOOD FORECASTING

Table 7.1. Structure of M4b₁ and M4b₂ models.

Model	Inputs	Output	MFs	No. parameters		
				Premise	Consequence	Total
M4b ₁	3	1	3	27	108	135
M4b ₂	3	1	3	27	108	135

The daily water levels at NongKhai, Thakhek, Pakse and Kratie and average rainfall for the sub-basins defined by the reaches NongKhai-Thakhek, Thakhek-Pakse, Pakse-Kratie for 10 wet seasons (1994 to 2003) were used for this study. The performance of on-line and off-line ANFIS was compared for two scenarios:

- (i) In the first scenario, it was assumed that a flood forecasting model needed to be setup but data was not available. Thus, at the end of year 1994, after 1 year of data was collected, the collected data was used to train the on-line and off-line ANFIS models. The on-line ANFIS model was also trained in the batch mode, to initialize the on-line model which was applied from 1995 onwards. For the off-line model, the model was retrained after each year, updating the model once a full year of data became available for training, and used in predictive mode for the following (full) year. Retraining was not required for the on-line model, since the on-line model is continuously updated at daily intervals, when additional input-output data pairs became available. Thus, the first scenario provides a comparison between on-line and off-line ANFIS models when data is limited at the initial stage of data collection.
- (ii) In the second scenario, 4 years' (1994 to 1997) data was used to train the on-line and off-line ANFIS models in batch mode. Data from 1998 to 2000 was used for model validation and the data from 2001 to 2003 was used to test the models. Large flood events were recorded in 2000 and 2002, providing a good test data set. In the second scenario, the off-line model is trained with sufficient data. Therefore, a comparison between the 2 models under such a

scenario investigates whether there is any advantage of the on-line model over the off-line model when sufficient training data is available.

7.4. Results and Discussion

7.4.1. Scenario 1

The comparisons of MAE between the on-line and off-line ANFIS models and Δ MAE (MAE of off-line model - MAE of on-line model) for 1-, 3-, and 5-lead-day water level forecasts at Thakhek, Pakse, and Kratie are shown in Figs. 7.2, 7.3, and 7.4, respectively. The errors of the off-line model for the first year of simulation (1995) are higher than the errors for the on-line model for all 1-, 3-, and 5-lead-day forecasts at all stations. This means the off-line model trains poorly when the data set of only one wet season was used for training the off-line model in comparison with the on-line model. When the off-line model was trained periodically after the end of each year, its performance improved. The results show that the MAE for the off-line model reduced significantly from 1995 to 1998 after which it reached an asymptotic value. This suggests the off-line model requires substantially more training data compared to the on-line model. The on-line model performed better than the off-line model for all cases. However, Δ MAE becomes less significant when more data is available for training; after 1998, Δ MAE between the models is small.

The MAE benchmarks adopted by Mekong River Commission (<http://ffw.mrcmekong.org/accuracy.htm>) are also included in the figures. Figure 7.2 shows that for Thakhek and Kratie, both models require approximately the same duration of training period before the MRC's benchmark for the MAE is met, up to 1996 and 1997 for Thakhek and Pakse, respectively. For Kratie, both models are unable to meet the benchmark set by MRC. Nevertheless, the advantage of the on-line model is clearly demonstrated since the MAE is always larger for the off-line model compared to the on-line model. Specifically, for Pakse, in the early years when data is scarce, the performance of the on-line model is significantly improved compared to the off-line model. For 3-lead-day water level forecasts at Thakhek,

Fig. 7.3 shows that the MAE of the on-line model is very close or within the benchmark MAE for predictions of water levels in 1997 and onwards. The off-line model requires an addition year of data for training (1994 – 1997) in order to achieve MRC's benchmark. For Pakse, the on-line model gives excellent results after 1996, whereas the off-line model achieves the benchmark from 1998 onwards. Although both models were not able to achieve the MRC benchmark; however, the MAE achieved by the on-line model is always smaller than the MAE for the off-line model. For 5-lead-day forecasts, the results from both on-line and off-line models are similar, except for 1995 predictions; nevertheless the MAE for both models are excessive. For Pakse, the MAE benchmark is met by the on-line model from 1997 onwards while the off-line model is able to achieve results close to the benchmark a year later. For Kratie, the benchmark was not achieved by both models; however the MAE for the on-line model is significantly smaller than the MAE for the off-line model up to 1997.

Figure 7.2. Comparison between on-line and off-line ANFIS models for 1-lead-day water level forecasts at (a) Thakhek, (b) Pakse, and (c) Kratie.

Figure 7.3. Comparison between on-line and off-line ANFIS models for 3-lead-day water level forecasts at (a) Thakhek, (b) Pakse, and (c) Kratie.

Figure 7.4. Comparison between on-line and off-line ANFIS models for 5-lead-day water level forecasts at (a) Thakhek, (b) Pakse, and (c) Kratie.

7.4.2. Scenario 2

It has been shown above that if an off-line ANFIS model was trained with approximately 4 years (1994 to 1997) of data, its overall performance approaches that of an on-line model. This section will present a more detailed comparison between on-line and off-line ANFIS models when 4 years of data are used for training (1994 to 1997). The validation and testing data sets chosen were water levels from 1998 to 2000 and 2001 to 2003, respectively. The comparison of Mean Absolute Error (MAE) of these two models is shown in Fig. 7.5.

Figure 7.5. Mean Absolute Error of M4b₁ and M4b₂ models for training, validation and test datasets at (a) Thakhek, (b) Pakse and (c) Kratie.

In Fig. 7.5, M4b₁ and M4b₂ models give the same results after the off-line training phase (recall that for the on-line model, this initial training phase is used to initialize the model parameters). For model M4b₁, the validation and test datasets were run without further updating of parameters, whereas the M4b₂ model was allowed to be trained continuously by the on-line learning method during the testing phase. It is clear that model M4b₂ achieves consistently lower MAE for the validation and test data sets. This demonstrates the on-line learning capability of model M4b₂ and shows the advantage of an on-line learning model. For the offline model, re-training of the model may be necessary in order to improve its performance.

7.4.3. Analysis of the Rising and Falling Phases of Flood Hydrographs

Further analysis was conducted to investigate the performance of the on-line model by comparing the rising and falling limbs of the simulated and measured water levels during actual flood events. Only the portions of the water level time series within the medium and higher regimes were considered in this analysis since accurate forecasts are especially important at the high levels. The medium, high, very high water levels at a station were defined in Table 4.3 in Chapter 4. Eight flood events recorded from 1994 to 2003 at Thakhek, Pakse and Kratie stations were chosen for this analysis. Table 7.2 (a) and (b) show details of the rising and falling limbs at Thakhek, respectively, and Fig. 7.6 shows the definitions of typical rising and falling phases of the water level time series for year 2000 (Events 4 and 5). Events 1, 2, and 3 were used for training, events 4 and 5 were for validation and events 6, 7 and 8 were used for testing.

CHAPTER 7: APPLICATION OF ON-LINE MODELS FOR FLOOD FORECASTING

Table 7.2. Data for (a) Rising and (b.) Falling limbs for flood events at Thakhek during the period 1993 to 2003.

(a.)

Events	Number of days	Duration		Water level		Average rate of increase (m/day)
		First day	Last day	Initial (m)	Peak (m)	
1	43	25/07/1995	05/09/1995	9.09	14.85	0.14
2	14	13/09/1996	26/09/1996	9.20	13.71	0.35
3	14	29/08/1997	11/09/1997	10.46	13.79	0.24
4	14	05/07/2000	18/07/2000	8.80	12.32	0.25
5	23	23/08/2000	14/09/2000	10.25	14.35	0.18
6	29	19/07/2001	16/08/2001	11.00	13.73	0.10
7	13	03/09/2001	15/09/2001	11.58	12.91	0.10
8	27	16/07/2002	11/08/2002	11.37	13.44	0.08

(b.)

Events	Number of days	Duration		Water level		Average rate of increase (m/day)
		First day	Last day	Initial (m)	End (m)	
1	31	06/09/1995	06/10/1995	14.82	9.00	-0.19
2	19	26/09/1996	14/10/1996	13.71	9.05	-0.25
3	13	12/09/1997	24/09/1997	13.70	9.00	-0.36
4	35	19/07/2000	22/08/2000	12.25	10.32	-0.06
5	22	15/09/2000	06/10/2000	14.32	8.83	-0.25
6	23	17/08/2001	02/09/2001	12.82	11.62	-0.05
7	15	16/09/2001	30/09/2001	12.72	9.70	-0.20
8	40	12/08/2002	20/09/2002	13.32	10.07	-0.08

Figure 7.6. Definition of the rising and falling limbs of flood events recorded at Thakhek. (a) Year 2000 from 01 Jun 2000 to 31 Oct 2000, and (b) Year 2001 from 01 Jun 2001 to 31 Oct 2001.

The MAE for the M4b₁ and M4b₂ models for the rising limb portion of the eight flood events are shown in Fig 7.7. The figure shows that M4b₂ model has consistently smaller MAE than M4b₁ model for 1-, 3- and 5-lead-day forecasts in almost all validation and testing events. The difference in the MAE (Δ MAE) (m), between the M4b₁ and M4b₂ models for the validation and testing flood events, varies according to the event, station and lead time of forecasts. The minimum and maximum values of Δ MAE between the models (MAE of M4b₁ - MAE M4b₂) for 1-, 3- and 5-lead-day forecasts for Thakhek, Pakse and Kratie are presented in Table 7.3. Table 7.3 shows that the M4b₂ model performed better than M4b₁ model in all cases with the minimum Δ MAE varying from 0.01 to 0.08 m. In addition, Table 7.3 shows that the greatest advantage of the M4b₂ model over the M4b₁ is for 3-lead-day forecasts since the maximum value of Δ MAE occurs consistently for 3-lead-days at all the stations considered. As shown in Fig. 7.7, for 5-lead-day forecasts, M4b₂ model performance deteriorates and the MAE for the M4b₂ model are not much smaller than that of the M4b₁ leading to a reduction in maximum value of Δ MAE.

Table 7.3. The range (min, max) of Δ MAE (m) between M4b₂ and M4b₁ model for the rising limb of the validation and test events.

Forecast lead days	Thakhek		Pakse		Kratie	
	min	max	min	max	min	max
1	0.01	0.13	0.01	0.04	0.03	0.15
3	0.01	<u>0.20</u>	0.05	<u>0.38</u>	0.08	<u>0.36</u>
5	0.06	0.16	0.02	0.30	0.05	0.21

Figure 7.7. Mean Absolute Error for M4b₁ and M4b₂ for the rising limbs of 8 flood events for (a) Thakhek, (b) Pakse and (c) Kratie.

In practice, flood events can be characterised by the rate at which stage changes (dH/dt) in the rising and falling phases of the flood hydrograph, and the ability to accurately predict the water level, especially in the rising phase, is important. Table 7.2 shows the average rate of change of water levels for the events considered where event 8 has the lowest rate of increase (0.08 m/day) while event 2 has the highest rate of increase (0.36 m/day). As a consequence, the performance of a flood forecasting model can vary due the differences in dH/dt between flood event. The

MAE and Δ MAE for the on-line and off-line models for the rising phase is plotted as a function of dH/dt in Fig. 7.8 which shows that the MAE for both models increases when dH/dt increases, which is to be expected. In addition, Δ MAE reduces when dH/dt increased. This indicates that the advantage of on-line learning diminishes when dH/dt increases. Thus, both on-line and off-line model forecast results are compromised when dH/dt is high. Although the parameters in the on-line model are continuously being updated; however, a long memory of up to 100 days (corresponding to a forgetting factor of 0.99) was required to obtain stable performance and reasonable results. This requirement compromised the results of the on-line model, since an inordinate amount of data, more than was helpful, had to be used to update the model. The specification of too long a memory when dH/dt is high for model training is disadvantageous since model training should be based on the more recent water levels (fewer rather than more data) as it is the most recent changes in water levels that would have a significant influence on forecast results.

It is also observed in Fig 7.8 that Δ MAE for 3-lead-day forecasts are higher than that for 1- and 5-lead-day forecasts (refer also to Table 7.3) meaning that the greatest improvement provided by the on-line model is for 3-lead-day forecasts. The superiority of the M4b₂ is not readily significant since the M4b₁ model is accurate for 1-lead-day forecasts. Forecasts for 5-lead days however appear to be a challenge for both models, due to the problems associated with timing errors in the forecast. This can be seen from plots of the water levels at Thakhek, Pakse, and Kratie simulated by the M4b₁ and M4b₂ models for 1-, 3-, and 5-lead-days forecasts, shown in Figure 7.9 (a), (b) and (c), respectively. Both models exhibit lag time errors, i.e., the simulated values are lagged to observed values, and the lag time error increases from 1- to 5-lead-days forecasts. Thus, while the MAE for the M4b₂ model is smaller, the improvement over the M4b₁ is not as significant. From this analysis, it is evident that the main advantage of the on-line model is in the 3-lead-day forecasts, where significant improvements over the M4b₁ model are obtained.

CHAPTER 7: APPLICATION OF ON-LINE MODELS FOR FLOOD FORECASTING

Figure 7.9. Comparison between observed and simulated water levels for (a.) 1-lead-day, (b.) 3-lead-days, and (c.) 5-lead-days forecasts.

CHAPTER 7: APPLICATION OF ON-LINE MODELS FOR FLOOD FORECASTING

The MAEs for the M4b₁ and M4b₂ models for the falling limb of the eight flood events under study are shown in Fig 7.10. The M4b₂ model gives smaller MAE than M4b₁ model for all forecast lead days. The range of Δ MAE between the M4b₂ and M4b₁ models is shown in Table 7.4. Similar to the rising phase, the largest value of Δ MAE was obtained for 3-lead-days forecasts.

Table 7.4. The range (min, max) of Δ MAE (m) between M4b₂ and M4b₁ models for the falling limb of the validation and test events.

Forecast lead days	Thakhek		Pakse		Kratie	
	min	max	min	max	min	max
1	0.02	0.06	0.00	0.08	0.03	0.15
3	0.01	<u>0.09</u>	0.02	<u>0.19</u>	0.11	<u>0.38</u>
5	0.02	0.08	0.01	0.17	0.05	0.21

As observed in Fig. 7.8, Fig. 7.11 shows that the MAEs for both models increase when dH/dt increases in the falling phase. In addition, the advantage of the on-line model diminishes when dH/dt is large for reasons explained previously. Further comparisons between the observed and water levels simulated by the M4b₁ and M4b₂ models for Thakhek, Pakse, and Kratie stations are shown in Figure 7.12 (a), (b) and (c) for 1-, 3-, and 5-lead-day forecasts, respectively. In general, the models are able to predict the falling phase better than the rising phase. This is because the falling phase usually occurs over a longer duration with a relatively lower rate of water level change than that of rising phase which usually occurs over a shorter duration (e.g., one or two weeks). When changes are not as rapid, both on-line and off-line models are able to capture the input-output relationship well. In addition, for the on-line model, the specification of a long memory is not as problematic as compared to the rising phase where the water level change is more rapid.

Figure 7.10. Mean Absolute Error for M4b₁ and M4b₂ for the falling limbs of 8 flood events for (a) Thakhek, (b) Pakse and (c) Kratie.

Figure 7.11. Evaluation of off-line and on-line models for falling limbs of validation and test flood events at (a) Thakhek, (b) Pakse, and (c) Kratie.

CHAPTER 7: APPLICATION OF ON-LINE MODELS FOR FLOOD FORECASTING

Figure 7.12. Comparison of water level time series at Thakhek, Pakse and Kratie for (a.) 1-lead-day, (b.) 3-lead-day, and (c.) 5-lead-day forecasting.

7.5 Conclusions

This chapter presented the comparison of model performance between on-line and offline ANFIS models. The results show that the off-line ANFIS model performed worse when there was a lack of training data. The performance of the off-line model improved when more training data became available. However, even when sufficient training data was available, it was shown that the on-line model was able to provide better results than the off-line model. It should however be noted that the on-line model was able to provide significantly improved results over the off-line model for 3-lead-days forecast. For 1-lead-day, the off-line model results were good and for 5-lead-days, lag time errors compromised on-line model results.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

The present study aimed to investigate the applicability of data-driven modelling for flood forecasting in a large river system, with the Lower Mekong adopted as a case study. The Adaptive Neuron-Fuzzy Inference System or ANFIS was chosen as it is a modern approach in Computational Intelligence for data-driven modelling. The following are conclusions obtained from this study:

1. The study has demonstrated the applicability of ANFIS for flood forecasting of the Lower Mekong. It was found that the model errors increased when the lead time of forecast increased from 1 to 5 days. The best agreement between forecast and measured water levels were obtained for Thakhek and the worst for Kratie. The ANFIS models provided acceptable levels of the MAE, when evaluated for whole flood season, for 1- to 3-lead-day water level forecasts since the MAE was less than the benchmark provided by the Mekong River Commission. However, when the MAE was computed for water levels during the rising phase, the MAE was found to be higher than the benchmark for certain flood events. Nevertheless, comparing between ANFIS and URBS models' results for the year 2009, the ANFIS model produced better results compared to the URBS model for 1- to 3-lead-day forecasts. However, ANFIS achieved only slight improvements in results compared to the URBS model for 4- and 5-lead-day forecasts.
2. The study highlights that the challenge for water level forecasting for long lead times remains. At long lead times, model errors are dominated by lag time errors with the predicted results leading the observed water levels. Thus, in the rising phase of a flood event water level forecasts are usually lower than the observed values and in the falling phase, forecasts are usually higher than the observed water level. Considerable improvements to 3-, 4- and 5-lead-day forecasts were obtained when ANFIS model results were updated by output updating procedures. Among the developed updating procedures, the partial

recursive output error prediction procedure, RAR_b , which uses the RAR model to approximate errors and uses the ANFIS₂ model results to initiate the error estimation process, gave the best results; this method was able to reduce the lag time errors of the ANFIS models for the falling limbs considerably, but only slightly for rising limb.

3. The input variables can play different roles in impacting ANFIS model performance. The choice of inputs to be used in forecast applications for different locations along the river needs to be carefully considered. For example, in addition to the water levels at the station of interest, the water level from an upstream station was sufficient for 1- and 3-lead day water level forecasts. The addition of rainfall information is not necessary for one- and three-lead-day forecasts. However, the inclusion of rainfall and an input may or may not be effective for longer lead time forecasts since it depends on the cross-correlation between rainfall and water level. It was found that rainfall inputs helped to reduce forecast errors for Thakhek for all low, medium, high and very high water level regimes significantly. On the other hand, significant improvements were obtained only for the medium water level regime at Kratie.
4. Off-line ANFIS models have parameters which are identified based on a given training dataset. Since the training dataset consists of data for a number of flood seasons, this may not be representative of the input-output relationship for a specific event. Thus, it is reasonable to assume that if model parameters are updated continuously when new data are presented, model performance will improve since the model is continuously trained using up-to-date information. The results from the on-line ANFIS model showed improvement in model performance. Although the results showed that the on-line training may not improve the accuracy of forecasts in the rising phase of flood events significantly, it can reduce forecast errors in the falling phase of flood events where the rate of change in water level is not as large.

The major contributions are as follows:

1. A comprehensive review of ANFIS model which is the current state-of-the-art computational intelligence inspired data driven technique, i.e. the hybrid of neural network and fuzzy inference systems, and applications of this model for hydrological forecasting including river flood forecasting.
2. The comparison between ANFIS results and the improvements of results over the current operational flood forecasting model for Mekong River Commission has shown that ANFIS can be used for real time flood forecasting in a large river system.
3. The importance of the relationship between hydrological characteristics and the inputs for ANFIS model has been demonstrated. This means that considerations of the flow contribution from upstream, sub-basins and rainfall have to be considered for individual stations along the main stream. Where the rainfall within a sub-basin provides a significant contribution to the total flow, the inclusion of rainfall inputs is advantageous in improving long lead time forecast (e.g. 5 days in the present study). In instances where rainfall contribution is not significant or where the terrain is steep, the lead time where accurate forecasts can be obtained is limited to the travel time of the flood wave from the upstream station to the station of interest.
4. The proposed output error correction procedure based on recursive auto-regressive (RAR) model outperforms the conventional method based on the auto-regressive (AR) model. Although this procedure was used to correct output errors of the ANFIS model, however, it would be equally applicable to correct output errors of other flood forecasting models.
5. The superiority of ANFIS using on-line learning method over the one using off-line learning method has been show. This highlights the potential for further studies aimed at improving the on-line learning algorithms for neuron-fuzzy systems for implementation in a real-time flood forecasting model.
6. Following the approach presented, ANFIS models can be setup to provide water level forecasts at all gauging stations along the lower Mekong. The Mekong River Commission may consider using ANFIS models in parallel independently from the models currently in use at the Flood Management and

Mitigation Center. Hence the ANFIS models can be alternative flood forecasting models for Mekong River Commission. Furthermore, we are currently collaborating with the Mekong River Commission to study ensemble forecasting method which explores how to archive more accurately forecasting by utilizing forecasts from multiple models.

Recommendations for further studies in future include the following:

1. The comparison between ANFIS and URBS's results in the present study was based on one year data only (i.e. year 2009). More hydrological data and results of the operational URBS model at RFMMC-MRC for years from 2010 to 2012 should be collected for additional modelling, comparison and analysis.
2. The present study has used an on-line learning algorithm for ANFIS. This algorithm allows parameters of ANFIS's rules to be updated when model received new data during its operation. However, the number of rules identified after training is fixed, the parameters of rules can be updated but no more would be added. Other algorithms such as Dynamic Evolving Neural-Fuzzy Inference System (DENFIS) proposed by Kasabov and Song (2002) has an on-line learning algorithm which allows the rules for the neural-fuzzy system to be indentified dynamically where new rules can be added to the system. The applicability of new online learning algorithms should be verified by comparing the performance of ANFIS, DENFIS and URBS models for years from 2009 to 2012.
3. The present study is limited to the Lower Mekong. Although the ANFIS models developed for the three locations of interest in the present study (i.e. Thakhek, Pakse, and Kratie) has shown the applicable of ANFIS when it is applied to water level forecasting at locations with different hydrological conditions, it would be more informative if other river systems also be taken into account. Therefore, additional application of ANFIS on other large river systems should be attempted.

PUBLICATIONS

1. Nguyen K T Phuoc and Chua H C Lloyd. 2011a. “Data-Driven Approach for Multistep Ahead Flood Forecasting for the Lower Mekong River” Proceedings of the 4th ASCE-EWRI International Perspective on Water Resources & Environment conference, National University of Singapore, Singapore, January 2011.
2. Nguyen K.T Phuoc, Chua H. C. Lloyd, 2011b. The Data-Driven Approach as an Operational Real-Time Flood Forecasting Model. Hydrological Processes (2011) DOI:10.1002/hyp.8347.

REFERENCES

- Abrahart RJ, 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: 2157-2172
- Abrahart, R. J., Heppenstall, A. J., & See, L. M. 2007. Timing error correction procedure applied to neural network rainfall-runoff modelling. *Hydrological Sciences Journal*, 52(3), 414-431.
- Antar MA, Ellassiouti I, Allam MN. 2006. Rainfall-runoff modelling using artificial neural networks technique: A Blue Nile catchment case study. *Hydrological Processes* 20: 1201-1216.
- Apirumanekul, C. 2006. Flood forecasting in the Mekong River Basin: an improvement plan for the flood forecasting system. *International Conference on Mekong Research for the people of the Mekong*, 18-21 October, Chiang Rei, Thailand.
- ASCE. 1993. Criteria for evaluation of watershed models. *J. Irrigation Drainage Eng.* 119(3): 429-442.
- ASCE Task Force Committee on Application of Artificial Neural Networks in Hydrology. 2000a. Artificial Neural Networks in Hydrology. Part I: Preliminary Concepts. *Journal of Hydrologic Engineering*, 5(2), 115 – 123.
- ASCE Task Force Committee on Application of Artificial Neural Networks in Hydrology. 2000b. Artificial Neural Networks in Hydrology. Part II: Hydrologic Applications. *Journal of Hydrologic Engineering*, 5(2), 124 – 137.
- Boughton. 2004. The Australian water balance model. *Environmental Modelling & Software*, Volume 19, Issue 10, October 2004, Pages 943-956.
- Box, G.E.P., Jenkins, G.M. 1976. Time Series Analysis: Forecasting and Control. Holden-Day: Oakland, CA

REFERENCES

- Bray, M., Han, D., 2004. Identification of support vector machines for runoff modelling. *Journal of Hydroinformatics* 6, 265-280.
- Broersen P.M.T. 2007. Error correction of rainfall-runoff models with the ARMAseI program. *IEEE Transactions on Instrumentation and Measurement* 56: 2212-2219.
- Butts, M. B., Payne, J.T., Kristensen, M. and Madsen, H. 2004. An evaluation of the impact of model structure on hydrological modelling uncertainty for streamflow prediction. *Journal of Hydrology*, Volume 298, Pages 242-266.
- Campolo M, Andreussi P, Soldati A.1999. River flood forecasting with a neural network model. *Water Resources Research* 35: 1191-1197
- Campolo M, Soldati A, Andreussi P. 2003. Artificial neural network approach to flood forecasting in the River Arno. *Hydrological Sciences Journal* 48: 381-398
- Carrol D.G (2007). The manual for URBS – A Rainfall Runoff Routing Model for Flood Forecasting & Design (<http://www.urbs.com.au/>)
- Chang FJ, Chiang YM, Chang LC. 2007. Multi-step-ahead neural networks for flood forecasting. *Hydrological Sciences Journal* 52: 114-130
- Chang LC, Chang FJ, Chiang YM. 2004. A two-step-ahead recurrent neural network for stream-flow forecasting. *Hydrological Processes* 18: 81-92
- Chau KW, Wu CL, Li YS. 2005. Comparison of several flood forecasting models in Yangtze River. *Journal of Hydrologic Engineering* 10: 485-491
- Chen SH, Lin YH, Chang LC, Chang FJ. 2006. The strategy of building a flood forecast model by neuro-fuzzy network. *Hydrological Processes* 20: 1525-1540
- Choy, K., Chen, C., 2003. Modelling of river discharges and rainfall using radial basis networks based on support vector regression. *International Journal of Systems Science* 34, 763-773
- Corani G, Guariso G. 2005. Coupling fuzzy modeling and neural networks for river flood prediction. *IEEE Transactions on Systems, Man and Cybernetics Part*

REFERENCES

C: Applications and Reviews 35: 382-390

- Cunge, J. A., 1969, On the Subject of a Flood Propagation Computation Method (Muskingum Method), *Journal of Hydraulic Research*, v. 7, no. 2, p. 205-230.
- Dawson CW, Wilby R. 1998. An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal* 43: 47-66
- Dawson CW, Harpham C, Wilby RL, Chen Y (2002) Evaluation of artificial neural network techniques for flow forecasting in the River Yangtze, China. *Hydrology and Earth System Sciences* 6: 619-626.
- de Vos, N. J., and T. H. M. Rientjes (2008), Correction of timing errors of artificial neural network rainfall-runoff models, in *Hydroinformatics in Practice: Computational Intelligence and Technological Developments in Water Applications*, edited by R. J. Abrahart, L. See, and D. Solomatine, Springer, New York.
- Drucker, H., Burges, C., Kaufman, L., Smola, A., and Vapnik, V. 1997. Support vector regression machines, in *Advances in Neural Information Processing System*, Vol.9, MIT Press, Cambridge, MA.
- Evensen, G., 1994. Sequential Data Assimilation With a Nonlinear Quasi-Geostrophic Model Using Monte-Carlo Methods to Forecast Error Statistics. *Journal of Geophysical Research*, 99:10143-10162.
- Feldman, A.D. 1994. Assessment of forecast technology for flood control operation in G.Rossi, N. Harmancioglu and V.Yejevich (eds), *Coping with Floods*, Kluwer Academic Publishers, Dordrecht, 445-458.
- Freeze, R. A. and R. L. Harlan. 1969. Blueprint for a physically-based, digitally-simulated hydrologic response model. *Journal of Hydrology* 9(3) 237-258.
- Fujita, M., M.-L. Zhu, T. Nakoa, and C. Ishi. 1992. An application of fuzzy set theory to runoff prediction. *Sixth IAHR International Symposium on Stochastic Hydraulics*, Int. Assoc. for Hydraul. Res., Taipei, Taiwan.

REFERENCES

- Goswami M, O'Connor KM, Bhattarai KP, Shamsedin AY. 2005. Assessing the performance of eight real-time updating models and procedures for the Brosna River. *Hydrology & Earth System Sciences*, 9(4), 394-411.
- G. C. Goodwin and K. S. Sin. 1984. Adaptive Filtering Prediction and Control. Englewood Cliffs, NJ: Prentice-Hall.
- Gelb, A., Ed., 1974: Applied Optimal Estimation. The MIT Press, 374 pp.
- Georgakakos, K.P., Seo, D.-J., Gupta, H., Schaake, J., Butts, M.B., 2004. Towards the characterization of streamflow simulation uncertainty through multimodel ensembles. *Journal of Hydrology* 298, 222–241.
- Gupta A, 2007. Introduction in *Large Rivers: Geomorphology and Management*, Edited by Avijit Gupta © 2007 John Wiley & Sons, Ltd., ISBN 978-0-470-84987-3.
- Haykin, S. (1994). Neural networks: a comprehensive foundation. Mac-Millan, New York.
- Hedlund Ulf, Chayanis Manusthiparom, Kittipong Jirayoot, Ian Campbell. 2005. Water Statistics in the Mekong River Basin. *IWG-Env, International Work Session on Water Statistics*, Vienna, June 20-22, 2005.
- Hsu, K., V. H. Gupta, and S. Sorroshian. 1995. Artificial neural network modeling of the rainfall-runoff process, *Water Resour. Res.*, 31(10), 2517– 2530.
- Hundecha, Y., A. Bardossy, and H.-W. Theisen. 2001. Development of a fuzzy logic based rainfall-runoff model, *Hydrological Sciences Journal*, 46(3), 363– 377.
- Jang, J.-S. R. 1993. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3), 665-685.
- Kalman, R.E., 1960. A New Approach to Linear Filtering and Prediction Problem. *Transactions of the ASME-Journal of Basic Engineering*, 82(series D):35-45
- Kanning, K., Pich, S., Pengel, B. 2008. Flood forecasting accuracy for the Mekong River Basin. *6th Annual Mekong Flood Forum “Integrated approaches and applicable systems for medium-term flood forecasting and early warning in*

REFERENCES

- the Mekong River Basin” Proceedings*, Phnom Penh, Cambodia, 27-28 May 2008, 298- 308.
- Kazama, S., Muto, Y., Nakatsuji, K., Inoue, K. 2002 “Study on the 2000 Flood in the Lower Mekong by Field Survey and Numerical Simulation” Proc. 13th Congress APD/IAHR, 534-539 (2002).
- Kitanidis, P. K., and R. L. Bras (1980), Real-Time Forecasting With a Conceptual Hydrologic Model 2. Applications and Results, *Water Resour. Res.*, 16(6), 1034–1044.
- Krzysztofowicz, R. 2001. The case for probabilistic forecasting in hydrology. *Journal of Hydrology*, 249, pp. 2-9.
- Lees, M., Young, P. C., Beven, K. J., Ferguson, S. & Burns, J. 1994. “An adaptive flood warning system for the River Nith at Dumfries”. In *River flood hydraulics* (ed. W. R. White & J. Watts), pp. 65-75. Wallingford: Institute of Hydrology.
- Lekkas Demetris F. 2002. Development and comparison of data-based flow forecasting methods. *PhD thesis. Imperial College of Science, Technology and Medicine, the University of London*.
- Lennart Ljung (1987), *System Identification: Theory for the User*, Prentice Hall PTR, Englewood Cliffs, New Jersey 07632, ISBN 0-13-881640-9.
- Liong SY, Lim WH, Kojiri T, Hori T. 2000. Advance flood forecasting for flood stricken Bangladesh with a fuzzy reasoning method. *Hydrological Processes* 14: 431-448
- Liong SY, Sivapragasam C. 2002. Flood stage forecasting with support vector machines. *Journal of the American Water Resources Association* 38: 173-186
- Madsen, H., and C. Skotner. 2005. "Adaptive State Updating in Real-Time River Flow Forecasting - a Combined Filtering and Error Forecasting Procedure. *Journal of Hydrology* 308.1-4: 302-12.
- Maier HR, Dandy GC. 2000. Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications.

REFERENCES

- Environmental Modelling and Software* 15: 101-124
- Maier HR, Jain A, Dandy GC, Sudheer KP. 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 and Software*
- May RJ, Maier HR, Dandy GC, Fernando TMKG. 2008. Non-linear variable selection for artificial neural networks using partial mutual information. *Environmental Modelling and Software* 23: 1312-1326
- Moore, R.J (1986). "Advances in Real-Time Forecasting Practice" Invited Paper, Symposium on Flood Warning Systems, Winter Meeting of the River Engineering Section, *The Institution of Water Engineers and Scientists* , 23 pp
- Moradkhani, H., Hsu, and K.-L., 2005. Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter. *Water Resources Research* 41, W05012.
- MRC. 2002a. International Expert Meeting on Early Warning for the Mekong river, 27 Febr – 1 March 2002. *Mekong River Commission Secretariat*, Phnom Penh, Cambodia.
- MRC. 2002b. Flood Management program, *Mekong River Commission*, Program document, Phnom Penh, Cambodia Nov. 2002.
- MRC. 2005. Overview of the Hydrology of the Mekong Basin. *Mekong River Commission*, Vientiane, 73 p.
- MRC. 2007. About Floods in the Mekong Basin in *Annual Mekong Flood Report 2006*, Mekong River Commission, Vientiane, 2007. 76pp. ISSN: 1728-3248.
- MRC. 2008a. The flood hydrology of the major tributaries in the Lower Mekong Basin in *Annual Mekong Flood Report 2007*, Mekong River Commission, Vientiane 2008, 86 pp.
- MRC. 2008b. Indication of accuracy of flood forecasting in the Lower Mekong River, <http://ffw.mrcmekong.org/accuracy.htm>
- MRC. 2009. System Performance Evaluation Report, The MRC Technical Task

REFERENCES

- Group for verification of the new MRC Mekong Flood Forecasting System (FEWS-URBS-ISIS), *Mekong River Commission Regional Flood Management and Mitigation Centre*, Phnom Penh, Cambodia October 2009.
- Nayak PC, Sudheer KP, Rangan DM, Ramasastri KS. 2004. A neuro-fuzzy computing technique for modeling hydrological time series. *Journal of Hydrology* 291: 52-66
- Nayak PC, Sudheer KP, Rangan DM, Ramasastri KS. 2005. Short-term flood forecasting with a neurofuzzy model. *Water Resources Research* 41: 1-16
- Nash, J. E. and J. V. Sutcliffe (1970), River flow forecasting through conceptual models part I — A discussion of principles, *Journal of Hydrology*, 10 (3), 282–290.
- Nguyen KT Phuoc, Chua HC Lloyd., 2011. “Data-driven approach for multistep ahead flood forecasting for the Lower Mekong River” Proceedings of the 4th ASCE-EWRI International Perspective on Water Resources & Environment conference, National University of Singapore, Singapore, January 2011.
- Nguyen K.T Phuoc, Chua H.C Lloyd, 2011. The Data-Driven Approach as an Operational Real-Time Flood Forecasting Model. *Hydrological Processes* (2011) DOI:10.1002/hyp.8347.
- Plate, E.J., 2006. A forecasting system for the lower Mekong. In: *Mekong Research for the People of the Mekong, Chiang Rei, Thailand*, October 2006, 18-21.
- Plate, E.J. 2007. Early warning and flood forecasting for large rivers with the lower Mekong as example, *Journal of Hydro-environment Research* 1 (2007), 80-94.
- Reichle, R., McLaughlin, D., Entekhabi, D., 2002. Hydrologic data assimilation with the ensemble Kalman filter. *Monthly Weather Review* 130, 103-114.
- See L, Openshaw S. 1999. Applying soft computing approaches to river level forecasting. *Hydrological Sciences Journal* 44: 763-778
- See L, Openshaw S. 2000. A hybrid multi-model approach to river level forecasting. *Hydrological Sciences Journal* 45: 523-536
-

REFERENCES

- Serban, P. and A. J. Askew. 1991. "Hydrological forecasting and updating procedures". IAHS (*International Association of Hydrological Sciences*) Publication, Vienna, Austria.
- Shamseldin, A.Y. 1997. Application of a neural network technique to rainfall-runoff modeling. *Journal of Hydrology*. 199(3-4), 272-294.
- Shamseldin, A.Y. and O'Connor, K.M. 1999. A real-time combination Method for the outputs of different rainfall-runoff models. *Hydrol. Sci. J.*, 44, 895–912.
- Shamseldin, A. Y. and K. M. O'Connor. 2001. "A Non-Linear Neural Network Technique for Updating of River Flow Forecasts." *Hydrology and Earth System Sciences* 5.4: 577-97.
- Shamseldin, A. Y. 2010. Artificial neural network model for river flow forecasting in a developing country. *Journal of Hydroinformatics* 12: 22-35
- Sivapragasam, C. and Liong, S., 2005. Flow categorization model for improving forecasting. *Nordic Hydrology* 36, 37-48.
- Sivapragasam, C., Liong, S. Y., Pasha, M., 2001. Rainfall and runoff forecasting with SSA-SVM approach. *Journal of Hydroinformatics*. 3, 141-152.
- Sompong Boonprasert. 2009. Status of MRC Hydro-meteorological Network for River Monitoring. 7th Annual Mekong Flood Forum (AMFF-7). 2009
- Sugawara, M., 1995. Tank model. In *Computer models of watershed hydrology*, V. P. Singh (Ed.), Water Resources Publications, 165-214.
- Dimitri P. Solomatine, Roland K. Price. 2004. Innovative Approaches To Flood Forecasting Using Data Driven And Hybrid Modelling. *The 6th International Conference on Hydroinformatics - Liong, Phoon & Babovic (eds) © 2004 World Scientific Publishing Company*, ISBN 981-238-787-0
- Sevat, E., and A. Dezetter. 1991. Selection of calibration objective functions in the context of rainfall-runoff modeling in a Sudanese savannah area. *Hydrological Sci. J.* 36(4): 307-330.
- Sudheer KP, Gosain AK, Ramasastri KS (2002) A data-driven algorithm for constructing artificial neural network rainfall-runoff models. *Hydrological*

REFERENCES

Processes 16: 1325-1330

- Takagi, T., & Sugeno, M. 1985. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15(1), 116-132.
- The U.S. Army Corps of Engineers, The Hydrologic Engineering Centers River Analysis System (HEC-RAS). [Online]. Available: www.hec.usace.army.mil/software/hecras/
- Toth E, Brath A. 2007. Multistep ahead streamflow forecasting: Role of calibration data in conceptual and neural network modeling. *Water Resources Research* 43
- Werner, M.G.F., Schellekens, J., Kwadijk, J.C.J., 2005. Flood Early Warning Systems for Hydrological (sub-) Catchments. *Encyclopedia of Hydrological Sciences*, Wiley, UK.
- World Meteorological Organization (WMO). 1992. "Simulated Real-time Intercomparison of Hydrological Models", *World Meteorological Organization, Operational Hydrology Report No.38*. WMO-No. 779. Geneva Switzerland.
- Xiong, L. H., A. Y. Shamseldin, and K. M. O'Connor. 2001. A nonlinear combination of the forecasts of rainfall-runoff models by the first order Takagi-Sugeno fuzzy system, *Journal of Hydrology*, 245(1– 4), 196–217.
- Xiong L, O'Connor KM (2002) Comparison of four updating models for real-time river flow forecasting. *Hydrological Science Journal*, 47: 4, 621-640.
- Yu, X., Liong, S. Y., Babovic, V., 2004. EC-SVM approach for real time hydrologic forecasting. *Journal of Hydroinformatics* 6, 209-223.
- Yu, P. S., Chen, S. T., Chang, I. F., 2006. Support vector regression for real-time flood state forecasting. *Journal of Hydrology* 328, 704-761.
- Zhu, M.-L., M. Fujita, N. Hashimoto, and M. Kudo. 1994. Long lead time forecast of runoff using fuzzy reasoning method, *J. Jpn. Soc. Hydrol. Water Resour.*, 7(2), 83– 89.