Rainfall-runoff modelling with neuro-fuzzy systems

Amin Talei

2013

Amin Talei. (2013). Rainfall-runoff modelling with neuro-fuzzy systems. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/54946>

<https://doi.org/10.32657/10356/54946>

NANYANG
TECHNOLOGICAL
UNIVERSITY

RAINFALL-RUNOFF MODELLING
WITH NEURO-FUZZY SYSTEMS

AMIN TALEI

SCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING

2013

RAINFALL-RUNOFF MODELLING WITH NEURO- FUZZY SYSTEMS

AMIN TALEI

**SCHOOL OF CIVIL AND ENVIRONMENTAL ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY**

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

2013

External Examiners:

1- Associate Professor Asaad Shamseldin, Department of Civil and Environmental Engineering, University of Auckland, New Zealand.

2- Associate Professor Robert Abrahart, School of Geography, University of Nottingham, UK.

*This thesis is dedicated to my beloved parents
who have supported me by all means since the
beginning of my life.*

ACKNOWLEDGMENTS

I would like to express my gratitude to all those who have helped me in one way or another to complete my PhD studies at NTU. I wish to thank Nanyang Technological University (NTU), Singapore and the DHI-NTU Water & Environmental Research Centre and Education Hub (DHI-NTU Centre), for giving me the financial support for my PhD study.

I am also deeply indebted to my supervisor, Assistant Professor Dr. Lloyd H. C. Chua from the Division of Environmental and Water Resources Engineering, CEE, NTU, whose knowledge, experience, encouragement, support, and suggestions helped me during my research and writing of this thesis.

I would like to express my gratitude to Associate Professor Dr. Quek Hiok Chai from the Division of Computer Science, School of Computer Engineering (SCE), NTU, whose inspiring words and lectures, support, and expertise in artificial intelligence were significantly helpful. I also wish to thank Prof Quek for the use of the MATLAB code for POPFNN.

I also want to thank the Public Utilities Board, Singapore for the use of the Kranji data, Dr. Tommy S. W. Wong for the use of his experimental catchment data, Assistant Professor Qin Xiaosheng from the Division of Environmental and water Resources Engineering, CEE, NTU for the use of the Heshui data, and Professor Per-Erik Jansson from the Department of Land and Water Resources Engineering, School of Architecture and the Built Environment, KTH, Sweden for the use of the Rönne data.

My good friends, Dr. Omid Geramifard, Dr. Reza Haghpanah, Dr. Amir Shakouri, and Dr. Reza Khosrow-Abadi, who helped me with coding in MATLAB. I would like to thank them for their considerable assistance. I also like to express my appreciation to Ms. Ng Yen Nie, Project Officer, CEE, NTU whose guidance on the use of SWMM was very helpful in my research. I also thank all of my colleagues in DHI-NTU Centre for their warm support and kind encouragement.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENT	i
TABLES OF CONTENTS	ii
ABSTRACT	vii
LIST OF FIGURES	ix
LIST OF TABLES	xiv
LIST OF SYMBOLS	xvii
CHAPTER 1	INTRODUCTION
	1
CHAPTER 2	LITERATURE REVIEW
	3
2.1	Introduction
	3
2.2	Input Selection
	8
2.3	Training Event Selection in an Event-based R-R Modeling
	10
2.4	Adaptability
	12
2.5	Interpretability
	14
2.6	Discussion
	16
CHAPTER 3	METHODOLOGY
	19
3.1	Introduction
	19
3.2	Fuzzy Logic Systems
	19
3.2.1	Introduction to fuzzy sets and fuzzy logic
	19
3.2.2	Fuzzy implication (If-Then rules)
	21
3.2.3	Fuzzy inference system
	22
3.2.4	Neuro-fuzzy systems
	25
3.2.5	Adaptive Network-Based Fuzzy Inference System (ANFIS)
	30
3.2.6	Over-fitting
	35
3.3	Physically-Based Models
	35
3.3.1	Kinematic Wave Model (KWM)
	35
3.3.2	Storm Water Management Model (SWMM)
	36

3.3.3	University of Regina Hydrologic Model (URHM)	37
3.4	Statistical Models	37
3.4.1	Naïve Model	37
3.4.2	Autoregressive Model with Exogenous Inputs (ARX)	37
3.5	Study Sites and Data Used	38
3.5.1	Experimental catchment (Catchment 1)	38
3.5.2	Kranji sub-catchment (Catchment 2)	39
3.5.3	Heshui catchment (Catchment 3)	40
3.5.4	Rönne catchment (Catchment 4)	41
3.6	Data Standardization	42
3.7	Data Smoothing	43
3.8	Model Performance Measures	44
3.9	Input Selection Methods	45
3.9.1	Introduction	45
3.9.2	Correlation Analysis	45
3.9.3	Mutual Information-Correlation Analysis	46
3.9.4	Stepwise Regression	48
CHAPTER 4	EVALUATION OF RAINFALL AND DISCHARGE INPUTS USED BY ANFIS IN RAINFALL-RUNOFF MODELING	49
4.1	Introduction	49
4.2	Input Candidate Selection	49
4.3	Training Event Selection	53
4.4	Choice of Membership Function (MF)	55
4.5	Over-fitting Issue	58
4.6	Models Development	60
4.7	Results and Discussion	66
4.7.1	Goodness-of-fit	66

	4.7.2	Peak Estimation	73
	4.7.3	Time shift error	75
	4.8	Models Comparison	79
	4.9	Conclusions	84
CHAPTER 5		TRAINING EVENT SELECTION FOR EVENT-BASED RAINFALL-RUNOFF MODELING BY ANFIS	85
	5.1	Introduction	85
	5.2	Influence of Hydrograph Shape on Event-Based Rainfall-Runoff Modelling	85
	5.2.1	Selection of Training Event Candidates	85
	5.2.2	Selection of Input Candidates	87
	5.2.3	ANFIS Models Development	88
	5.2.4	SWMM Models Calibration	90
	5.2.5	Results and Discussion	91
	5.3	Influence of Lag Time on Event-Based Rainfall-Runoff Modelling	98
	5.3.1	Lag time and selection of training events	98
	5.3.2	Selection of inputs	100
	5.3.3	Results and discussions	101
		5.3.3.1 Effect of lag time	101
		5.3.3.2 Model development based on lag time	104
	5.4	Extension of Findings on Catchment 1	111
	5.5	Conclusions	112
CHAPTER 6		RAINFALL-RUNOFF MODELING USING A TAKAGI-SUGENO NEURO-FUZZY MODEL WITH LOCAL LEARNING	115
	6.1	Introduction	115
	6.2	Dynamic Evolving Neural Fuzzy Inference System (DENFIS)	116

6.2.1	Evolving Clustering Method (ECM)	116
6.2.2	Learning process in DENFIS	121
6.3	Methodology	122
6.3.1	Study Sites and Data Used	122
6.3.2	Physically-based models used	123
6.3.3	Input data selection and model development	123
6.4	Results and Discussions	125
6.5	Effect of Data Length on DENFIS Model Training	141
6.6	Effect of Data Sequence when Training Online Models	149
6.7	Real Time DENFIS (RT-DENFIS)	159
6.7.1	Introduction	159
6.7.2	Application of RT-DENFIS in Discharge Forecasting	160
6.8	Conclusions	166
CHAPTER 7	SIMILARITY-BASED FUZZY NEURAL NETWORK FOR RAINFALL-RUNOFF MODELING	168
7.1	Introduction	168
7.2	Approximate Analogical Reasoning Schema (AARS)	168
7.2.1	Similarity measures	169
7.2.2	Modification functions (ModF)	171
7.3	POPFNN-AARS	172
7.3.1	Structure	172
7.3.2	Learning algorithm	177
7.3.3	Over-fitting	181
7.4	Methodology	182
7.4.1	Study sites and data used	182
7.4.2	Input selection and model development	182
7.5	Results and Discussion	185
7.6	Conclusions	199
CHAPTER 8	DISCUSSION	201

8.1	Input Selection in Event-based R-R Modelling	201
8.2	Training Event selection in Event-based R-R Modelling	202
8.3	Adaptable Neuro-Fuzzy Systems	202
8.4	Interpretable Neuro-Fuzzy Systems	204
CHPATER 9	CONCLUSIONS	206
REFERENCES		209
APPENDIX A		219
APPENDIX B		220
LIST OF PUBLICATIONS		223

ABSTARCT:

Intelligent computing tools based on Fuzzy Logic and Artificial Neural Networks (ANN) have been used in hydrological modeling with promising results. A new approach of combining these two powerful artificial intelligence or AI tools, known as Neuro-Fuzzy Systems (NFS), has increasingly attracted researchers in various fields. Although many studies have been carried out using this approach in rainfall-runoff (R-R) applications, relatively few studies have been undertaken to evaluate the fundamental behavior of such models. This study employed NFS for R-R modeling and attempts to address issues related to (i) input selection for NFS in event-based R-R modeling, (ii) Training event selection for NFS in event-based R-R modeling, (iii) adaptability in NFS for R-R modeling, and (iv) interpretability in NFS for R-R modeling.

ANFIS was employed for event-based rainfall-runoff modeling and investigations on input selection for ANFIS were conducted by using three input selection methods known correlation analysis (CA), mutual information-correlation analysis (MICA), and stepwise regression (SR). Results revealed that using non-sequential rainfall antecedents are better choices compared to sequential or pruned sequential rainfall antecedents and among the three input selection methods used, MICA is more reliable. The study on training event selection showed that the hydrograph shape is an important consideration in choosing training events. Events with a single peak in the hydrograph were found to be better choices compared to events with multiple peaks for training the ANFIS model. In addition, the lag time was found to be an important factor to be considered in training event selection as well. Based on this study, it is recommended that events with lag times close to the median lag time of the historical events can be chosen as events to be used for training the ANFIS model. Lastly, ANFIS results were found to be comparable with results obtained from the Storm Water Management Model (SWMM) and superior to that of ARX model.

A neuro-fuzzy system which uses online learning known as Dynamic Evolving Neural Fuzzy Inference System (DENFIS) was introduced for continuous and event-based rainfall-runoff modeling in different catchment sizes. The results showed that DENFIS was able to capture the rainfall-runoff relationship well when tested for 3 catchments with different sizes. DENFIS results were found comparable when compared with the results obtained from Kinematic Wave Model (KWM), SWMM, and the University of Regina Hydrologic Model (URHM). DENFIS showed superior results to the one obtained by ARX model. Although the improvement in results obtained by the local learning algorithm used in

DENFIS over the global offline learning algorithm used in ANFIS were modest, DENFIS had a shorter training time compared to ANFIS. This is because the local learning uses a one pass algorithm, compared to global learning which requires several epochs for training. DENFIS was considered a better model in finding the proper number of membership functions compared to ANFIS, since there is no need to find the suitable number of membership functions by trial. A sensitivity analysis revealed that DENFIS shows less sensitivity to the length of training data while ANFIS was found to be sensitive to the length of the training data used. Significantly, DENFIS required shorter length of training data compared to ANFIS. Lastly, DENFIS was found to be sensitive to the sequence of training data while ANFIS was not. A real time version of DENFIS (RT-DENFIS) which has real time learning capabilities was developed in this study and compared with ANFIS that employs an offline learning algorithm. The results revealed that an ANFIS model which employs batch learning algorithm trained with historical data needed to be retrained periodically in order to maintain reasonably good predictions. RT-DENFIS, however, does not need retraining since the model is constantly being updated when new data is available.

A similarity-based neuro fuzzy system known as Pseudo Outer Product Fuzzy Neural Network with Approximate Analogical Reasoning Schema (POPFNN-AARS) was introduced and employed for both event-based and continuous R-R modeling for three different catchment sizes. The results obtained from the POPFNN model were comparable with the results obtained from the physically-based models KWM, SWMM, and URHM. Results obtained by POPFNN were also superior to that of ARX model. POPFNN results were also comparable with the results obtained from an ANFIS model. Although the improvement in results obtained by the rule pruning mechanism used in the POPFNN learning algorithm over ANFIS which has a fixed number of rules was modest, POPFNN has a more flexible rule structure with an optimal number of rules to better capture the association between input and output. Moreover, the Mamdani-type FIS used in POPFNN was found to be able to provide a linguistic understanding for output variable in the consequent part of the rules while the Takagi-Sugeno-type FIS used in ANFIS specifies the rule consequents as quantitative or crisp values. This linguistic nature of the Mamdani-type FIS could be advantageous if a qualitative understanding for the output variable is necessary.

LIST OF FIGURES

Figure 3.1: Typical membership functions for a fuzzy set. (Left: a smooth curve function. Right: a piecewise linear function.)

Figure 3.2: Complement (NOT) of a fuzzy set.

Figure 3.3: Union (OR) of two fuzzy sets.

Figure 3.4: Intersection (AND) of two fuzzy sets.

Figure 3.5: Basic structure of a fuzzy inference system.

Figure 3.6: Mamdani's fuzzy inference system.

Figure 3.7: Takagi-Sugeno fuzzy inference system.

Figure 3.8: Cooperative neuro-fuzzy system type 1 (source: Karray and De Silva, 2004).

Figure 3.9: Cooperative neuro-fuzzy system type 2 (source: Karray and De Silva, 2004).

Figure 3.10: Cooperative neuro-fuzzy system type 3 (source: Karray and De Silva, 2004).

Figure 3.11: Cooperative neuro-fuzzy system type 4 (source: Karray and De Silva, 2004).

Figure 3.12: Concurrent neuro-fuzzy system (source: Karray and De Silva, 2004).

Figure 3.13: A typical structure for an Adaptive Network-based Fuzzy Inference system (ANFIS).

Figure 3.14: Pre-defined membership functions in Fuzzy Logic MATLAB Toolbox: (a) trimf, (b) trapmf, (c) gaussmf, (d) gaussmf2, (e) gbellmf, (f) sigmf, (g) dsigmf, (h) psigmf, (i) zmf, (j) smf, and (k) pimf. Source: Fuzzy Logic MATLAB Toolbox (MATLAB 2011b).

Figure 3.15: Schematic diagram of the experimental station.

Figure 3.16: Schematic map of Kranji sub-catchment.

Figure 3.17: Schematic map of Heshui watershed.

Figure 3.18: Schematic map of Klippan_2, the sub-catchment of Rönne watershed.

Figure 4.1: Cross-correlation results between discharge at present time $Q(t)$ and rainfall at different lead times.

Figure 4.2: Comparison of ANFIS models' performance in terms of (a) CE , (b) r^2 , (c) $RMSE$, (d) MAE , and (e) RPE where different types and number of MFs are used.

Figure 4.3: Comparison of training versus verification performances due to the change in number of MFs used: (a)-(e) *tri* MF, and (f)-(j) *gauss* MF.

Figure 4.4: ANFIS performance during training and verification in terms of (a) CE , (b) r^2 , (c) $RMSE$, (d) MAE , and (e) RPE where number of epochs was varying.

Figure 4.5: Auto-correlation analysis for the discharge time series.

Figure 4.6: Partial Auto-correlation analysis for the discharge time series.

Figure 4.7: Average CE values for (a) Model 9 and (b) Model 9* for simulating discharge predictions from $Q(t)$ to $Q(t+8)$. Inputs used for discharge predictions are shown below the data points. (Training Event: Event 3; Verification Events: Events 5, 7, and 8).

Figure 4.8: r^2 values of the 9 events for (a) $Q(t)$, (b) $Q(t+1)$, (c) $Q(t+2)$, (d) $Q(t+4)$, and (e) $Q(t+8)$.

Figure 4.9: MAE values of the 9 events for (a) $Q(t)$, (b) $Q(t+1)$, (c) $Q(t+2)$, (d) $Q(t+4)$, and (e) $Q(t+8)$.

Figure 4.10: RPE values of the 9 events for (a) $Q(t)$, (b) $Q(t+1)$, (c) $Q(t+2)$, (d) $Q(t+4)$, and (e) $Q(t+8)$.

Figure 4.11: Variation of CE in response to shifts of the Model 5-simulated $Q(t+8)$ discharge for Event 10.

Figure 4.12: $\bar{\delta}$ values in discharge predictions for all nine events simulated by models (a) 1, (b) 5, (c) 9, (d) 1*, (e) 5*, and (f) 9*.

Figure 4.13: Plots of observed discharge versus the best and the worst simulated hydrographs of Event 2 for (a) $Q(t)$, (b) $Q(t+1)$, (c) $Q(t+2)$, (d) $Q(t+4)$, and (e) $Q(t+8)$.

Figure 5.1: Typical time series of rainfall and discharge for (a) Group 1, (b) Group 2, (c) Group 3 events in Catchment 2.

Figure 5.2: Events distribution in different intervals of CE for proposed models.

Figure 5.3: Scatter plot of *CE* values for events simulated by SWMM1 and ANFIS2.

Figure 5.4: Observed and simulated hydrographs of Event 30 Dec 04 (*CE* for SWMM1 and ANFIS2 are 0.466 and 0.813, respectively).

Figure 5.5: Observed and simulated hydrographs of Event 05 Jan 06 (*CE* values for SWMM1 and ANFIS2 are 0.843 and 0.462, respectively).

Figure 5.6: Scatter plot of *RPE* values for events simulated by SWMM1 and ANFIS2.

Figure 5.7: Scatter plot of *CE* versus *RPE* for events simulated by SWMM1 and ANFIS2: (a) Group 1, (b) Group 2, and (c) Group 3.

Figure 5.8: Schematic representation of lag time based on two different definitions.

Figure 5.9: Distribution of 62 events over different lag time.

Figure 5.10: *CE* for different models used to simulate $Q(t)$ for events of group (a) G5, (b) G6, (c) G7, and (d) G8.

Figure 5.11: r^2 for different models used to simulate $Q(t)$ for events of group (a) G5, (b) G6, (c) G7, and (d) G8.

Figure 5.12: *RMSE* for different models used to simulate $Q(t)$ for events of group (a) G5, (b) G6, (c) G7, and (d) G8.

Figure 5.13: Time series of rainfall and discharge used as training events for ANFIS1: (a) 06 May 05, (b) 16 May 05, (c) 26 Jul 05, (d) 16 Oct 05, and (e) 24 Dec 05.

Figure 5.14: Time series of rainfall and discharge used as training events for ANFIS2: (a) 19 Jul 05, (b) 30 Jul 05, (c) 20 Aug 05, (d) 16 May 05, (e) 05 Oct 05, (f) 17 Jan 06.

Figure 5.15: Time series of rainfall and discharge used as training events for ANFIS3: (a) 29 Jul 06, (b) 15 Apr 06, (c) 19 Jul 05, (d) 30 Jul 05, (e) 24 Dec 05, (f) 05 Oct 05, and (g) 05 Sep 06.

Figure 5.16: (a) *CE*, (b) r^2 , (c) *RMSE*, and (d) *RPE* for ANFIS1, ANFIS2, ANFIS3a, ANFIS3b, ANFIS3c models.

Figure 6.1: Schematic of ECM clustering procedure using with samples x_1 to x_9 in 2-D space.
(a) The first input, x_1 is adopted as the first cluster C_1^0 . (b) x_2 : updates cluster C_1^0

to C_1^1 ; x_3 : creates a new cluster C_2^0 ; x_4 : belongs to cluster C_1^1 thus no change is required. (c) x_5 : updates cluster C_1^1 to C_1^2 ; x_6 : belongs to cluster C_1^2 thus no change is required; x_7 : updates cluster C_2^0 to C_2^1 ; x_8 : creates a new cluster C_3^0 . (d) x_9 : updates cluster C_1^2 to C_1^3 . (Figure adapted from Kasabov and Song, 2002)

Figure 6.2: Schematic illustration of ECM algorithm.

Figure 6.3: Comparison between different models used for Catchment 1. (a) CE , (b) r^2 , (c) $RMSE$, (d) MAE , and (e) RPE .

Figure 6.4: Hyetograph, observed and simulated hydrograph for events simulated by DENFIS and KWM in Catchment 1.

Figure 6.5: Comparison between different models used for Catchment 2. (a) CE , (b) r^2 , (c) $RMSE$, (d) MAE , and (e) RPE .

Figure 6.6: Hyetograph, observed and simulated hydrograph for median events simulated by DENFIS and SWMM for Catchment 2.

Figure 6.7: Comparison of RPE values for peak discharges > 100 m³/s for Catchment 3.

Figure 6.8: Measured rainfall, observed discharge, and simulated discharges by DENFIS and URHM models for Catchment 3.

Figure 6.9: Rule creation as a function of input $Q(t-1)$.

Figure 6.10: Combination between 4 different input variables for data points which create a new rule.

Figure 6.11: Rainfall and discharge time series for the years: (a) 1997, (b) 1998, (c) 1999, (d) 2000, and (e) 2002.

Figure 6.12: Training and testing CE and r^2 values for different training data length: (a) ANFIS and (b) DENFIS models.

Figure 6.13: Measured rainfall, observed and simulated discharges by ANFIS and DENFIS models for Catchment 3 trained using training data length of (a) 1-year, (b) 2-years, (c) 3-years, and (d) 4-years.

Figure 6.14: Rainfall and discharge time series for years with (a) 1996 - low (L), (b) 1999 - medium (M), and (c) 1998 - high (H) flows.

Figure 6.15: Measured rainfall, observed discharge, and simulated discharge for ANFIS compared with DENFIS models trained with (a) LMH, (b) LHM, (c) MLH, (d) MHL, (e) HLM, and (f) HML training datasets for Catchment 3.

Figure 6.16: CE , r^2 , $RMSE$, and MAE values obtained by ANFIS and DENFIS models trained with data with six different for sequences.

Figure 6.17: Development of RT-DENFIS model.

Figure 6.18: Performance of ANFIS model for Scenario (1) and RT-DENFIS for Scenario (2) in simulating $Q(t+1)$ in terms of: (a) CE and (b) MAE .

Figure 6.19: Performance of ANFIS model for Scenario (3) and RT-DENFIS for Scenario (2) in simulating $Q(t+1)$ in terms of: (a) CE and (b) MAE .

Figure 7.1: Schematic process of approximate analogical reasoning schema in fuzzy inference.

Figure 7.2: Different forms of modification functions: (a) Expansion form; (b) Reduction form.

Figure 7.3: Structure of singleton fuzzifier POP-FNN.

Figure 7.4: Sample POP-FNN with two inputs and one output linguistic variable.

Figure 7.5: Schematic process of identifying rules in POP learning algorithm.

Figure 7.6: Comparison between different models used for Catchment 1. (a) CE , (b) r^2 , (c) $RMSE$, (d) MAE , and (e) RPE .

Figure 7.7: Hyetograph, observed and simulated hydrograph for events simulated by POPFNN and KWM in Catchment 1.

Figure 7.8: Comparison between different models used for Catchment 2. (a) CE , (b) r^2 , (c) $RMSE$, (d) MAE , and (e) RPE .

Figure 7.9: Hyetograph, observed and simulated hydrograph for median events simulated by POPFNN and SWMM for Catchment 2.

Figure 7.10: Comparison of RPE values for peak discharges $> 100 \text{ m}^3/\text{s}$ for Catchment 3.

Figure 7.11: Measured rainfall, observed discharge, and simulated discharges by POPFNN and URHM models for Catchment 3.

LIST OF TABLES

Table 3.1: Linguistic and precise hybrid neuro-fuzzy systems.

Table 4.1: Sequential and pruned sequential rainfall input time series selected by cross-correlation analysis for Catchment 1.

Table 4.2: Input candidates selected based on mutual information-correlation analysis for Catchment 1.

Table 4.3: Verification results obtained by ANFIS models trained with different training sets in Catchment 1.

Table 4.4: Verification results for ANFIS models with different input candidates.

Table 4.5: Details of the models chosen for comparisons in Catchment 1.

Table 4.6: Rainfall inputs used for prediction of $Q(t)$ to $Q(t+8)$ in Model 9.

Table 4.7: Ranking of the 7 models of Catchment 1 based on the proposed ranking procedure.

Table 4.8: Comparison between Naïve, ARX, and ANFIS models in Catchment 1.

Table 5.1: Selected training/verification events in ANFIS2 model in Catchment 2.

Table 5.2: Input candidates selected based on CA, MICA, and SR methods for ANFIS1 and ANFIS2 models in Catchment2.

Table 5.3: Verification results obtained by ANFIS2 trained with different training sets in Catchment 2.

Table 5.4: SWMM1 calibration: Estimated parameters and performance.

Table 5.5: SWMM2 calibration: Estimated parameters and performance.

Table 5.6: Average values of evaluation metrics over 66 events of Catchment 2 simulated by different models.

Table 5.7: Training events used for each group.

Table 5.8: Optimal combination of two rainfall inputs for models used in simulating $Q(t)$.

Table 5.9: Calibration events used in each ANFIS3 sub-models.

Table 5.10: Average values of CE , r^2 , $RMSE$, and RPE for 62 events of Catchment 2 obtained by different models.

Table 5.11: Selected training/verification events for Catchment1.

Table 5.12: Verification results obtained by ANFIS model trained with different training sets in Catchment 1.

Table 6.1: Events and inputs used for DENFIS training in Catchments 1 and 2.

Table 6.2: Details of data and inputs used for DENFIS training in Catchment 3.

Table 6.3: Comparison of DENFIS, ANFIS, KWM, and ARX models for Catchment 1.

Table 6.4: Comparison of DENFIS, ANFIS, SWMM, and ARX models for Catchment 2.

Table 6.5: Comparison of CE , r^2 , MAE , and $RMSE$ values for DENFIS, ANFIS, and URHM models for Catchment 3.

Table 6.6: Evaluating the training process in DENFIS and ANFIS.

Table 6.7: Creating and updating the rules in DENFIS model for normalized data in Catchment 3.

Table 6.8: The best 1, 2, 3, 4-years training data for ANFIS and DENFIS.

Table 6.9: Number of rules created in ANFIS and DENFIS models as a function of training data sequencing.

Table 6.10: Comparison of CE , r^2 , $RMSE$, and MAE values for RT-DENFIS, ANFIS, ARX, and Naïve models in predicting $Q(t+1)$, $Q(t+2)$, and $Q(t+3)$.

Table 7.1: Events and inputs used for POPFNN training in Catchments 1 and 2.

Table 7.2: Details of data and inputs used for POPFNN training in Catchment 3.

Table 7.3: Parameters of POPFNN model in Catchments 1, 2, and 3.

Table 7.4: Comparison of POPFNN, ANFIS, KWM, and ARX models for Catchment 1.

Table 7.5: Comparison of POPFNN, ANFIS, SWMM, and ARX models for Catchment 2.

Table 7.6: Comparison of CE , r^2 , MAE , and $RMSE$ values for POPFNN, ANFIS, URHM, and ARX models for Catchment 3.

Table 7.7: Extracted rules for Catchment 1 from POPFNN model (L and H represent low and high values, respectively).

Table 7.8: Extracted rules for Catchment 1 from ANFIS model with 2 membership functions for inputs and linear function for output (L and H represent low and high values, respectively).

Table 7.9: Extracted rules for Catchment 2 from POPFNN model (L and H represent low and high values, respectively).

Table 7.10: Extracted rules for Catchment 2 from ANFIS model with 2 membership functions for inputs and linear function for output (L and H represent low and high values, respectively).

Table 7.11: Extracted rules for Catchment 3 from POPFNN model (L, M, and H represent low, medium, and high values, respectively).

Table 7.12: Extracted rules for Catchment 3 from ANFIS model (L and H represent low and high values, respectively).

LIST OF SYMBOLS

δ	time shift error
μ	membership function
σ	width of the Gaussian membership function
τ	threshold of fuzzy rule base in POPFNN-AARS
φ	infiltration index
Δ_{∞}	Kaufman and Gupta Measure (∞)
Δ_*	Kaufman and Gupta Measure (*)
a_{ij}	penalty coefficient
c	centre of Gaussian membership function
d_2	Euclidian distance
$e(t)$	true error term in ARX model
f	net input
f_0	infiltration rate at the start of rainfall
f_c	ultimate infiltration rate
i_n	net rainfall
k	decay constant
m	smallest number of initial rules to be created in DENFIS
n	Manning's roughness coefficient
n_0	number of the first group of data pairs used to initialize DENFIS
n_a	number of past outputs in ARX model
n_b	nuber of past inputs in ARX model
n_{i-o}	number of input-output pattern in Akaike and Bayesian in formation criteria
n_k	delay associated with each input in ARX model
n_p	number of model parameters in Akaike and Bayesian in formation criteria
o	net output

q	number of input data points
q_{∞}	Hausdorff Measure (∞)
q_*	Hausdorff Measure (*)
t	time
t_l	lead time
t_m	the most correlated antecedent rainfall
x_n	standardized data
y	flow depth
C	covariance matrix
C_i	cluster
C_{xx}	variance of variable x
C_{xy}	between-sets covariance of variables x and y
C_c	cluster centre
D	distance from a cluster centre
D_p	depression storage for pervious sub-areas
D_{thr}	threshold value
I	independence
IL	input lable
Imp	impervious percentage
MI	mutual information
N_{imp}	Manning's roughness coefficient for impervious sub-areas
N_{per}	Manning's roughness coefficient for pervious sub-areas
OL	output label
Q	observed discharge or runoff
$\hat{Q}$	simulated discharge
$\overline{Q}$	average observed discharge

$\tilde{Q}$	average simulated discharge
Q_p	observed peak discharge
$\hat{Q}_p$	simulated peak discharge
$R(t)$	rainfall at time t
R	cluster radius
R_j	input candidate rank
S	overland slope
S_0	characteristic slope
S_4	Disconsistency measure
T	temperature
T_i	number of labels for node i
W	characteristic width

CHAPTER 1 INTRODUCTION

One of the most important problems in hydrology is the modeling of the rainfall–runoff (R-R) process. This is an important subject because of its vast applications in different hydrologic problems like flood forecasting, design of hydraulic structures, water quality modeling, and urban planning and management. Rainfall-runoff is a process which is highly affected by a variety of non-linear factors like rainfall, watershed characteristics, soil properties, etc. However, any effort to model the rainfall–runoff relationship would be confronted with difficulties since this process is highly nonlinear, time-varying, spatially distributed, and stochastic. In addition, deficiencies in data due to the unavailability of data, poor quality of data, etc. present a major problem in rainfall–runoff modeling.

To-date many methods and approaches have been introduced to model the rainfall–runoff relationship. These methods can be categorized into two main groups: physically-based models and system theoretic models. Physically-based models are designed to approximate the general internal sub-processes and physical mechanisms which govern the hydrologic cycle. They usually incorporate simplified forms of physical laws and are generally non-linear, time-varying, and deterministic, with parameters that are representative of watershed characteristics. Some examples for this group are the Storm Water Management Model (SWMM), the flood hydrograph package of Hydrologic Engineering Center of US Army Corps of Engineers (HEC-1), and the Sacramento Soil Moisture Accounting Model (SAC-SMA). Although physically-based models help us in understanding the physics of hydrological processes, they require sophisticated mathematical tools, and usually require significant user expertise.

On the other hand, system theoretic or black-box models apply a different approach to identify a direct mapping between rainfall and runoff, without the need for a detailed consideration of the physical processes. Linear time series models like Autoregressive Moving Average with Exogenous Inputs (ARMAX) and other linear and non-linear regression models, Artificial Neural Networks (ANN), and neuro-fuzzy systems (NFS) are examples of this group. Although these kinds of models are fast and their results are often comparable with physically-based models, they do not give us any information about the hydrologic process in the system.

In the present study, a literature review on the application of ANNs in rainfall-runoff modelling is provided in Chapter 2. Moreover, the gradual shift from completely black-box models such as ANN towards semantic-based models like NFS is discussed. The review of the studies in which ANN and NFS have been used in R-R modelling highlights some important topics in this field of research, including issues on model calibration for event-based R-R modelling such as input selection and training event selection. In addition, model adaptability and interpretability are also pertinent factors for these types of models, which are still not addressed sufficiently in literature. Therefore, this study aims to investigate the afore-mentioned issues in application of NFS in R-R modelling.

A general literature review on fuzzy logic and neuro-fuzzy systems and also a general briefing on the physically-based and regression-based R-R models used as bench mark for comparison are provided in Chapter 3, which includes detailed information on study sites, data used, data pre-processing methods, performance evaluation metrics, and input selection methods used in this study. Chapter 4 of this study is focused on the evaluation of rainfall and discharge inputs used by Adaptive Network-based fuzzy Inference System (ANFIS) in event-based rainfall-runoff modeling. The effect of hydrograph shape and lag time on training event selection for event-based rainfall-runoff modelling by ANFIS are discussed in Chapter 5. To study the adaptability of NFS, a Takagi-Sugeno neuro-fuzzy model with local learning known as Dynamic Evolving Neural-Fuzzy Inference System (DENFIS) (Kasabov and Song, 2002) and its application for rainfall-runoff modelling in three different catchment sizes are discussed in Chapter 6. In addition, a Real-Time DENFIS (RT-DENFIS) which is a self-learning model is proposed and its application for discharge forecasting is discussed in Chapter 6. A similarity-based fuzzy neural network (Quek and Zhou, 1999) is introduced in Chapter 7 to assess the capabilities of a more interpretable NFS model for rainfall-runoff modelling. Chapter 8 provides a general evaluation and interpretation of the findings presented in and also discusses the limitations of this study. Lastly, conclusions are summarized in Chapter 9.

CHAPTER 2 LITERATURE REVIEW

2.1. Introduction

Artificial neural networks, as generalizations of mathematical models of human cognition or neural biology, are massively parallel distributed processors made up of simple processing units known as neurons which are capable of storing experiential knowledge and making it available for use (Haykin, 1999). Mathematically, an ANN is often viewed as a universal approximator. It has the ability to identify the relationship from given patterns and solve large-scale complex problems such as pattern recognition, non-linear modeling, classification, association, and control. The early applications of ANNs in rainfall-runoff modeling were by Halff et al. (1993) and Hjelmfelt and Wang (1993). Later, efforts were made to have a clearer understanding of ANN capabilities in R-R modeling and assess the factors influencing their performance. For example, Chiang et al., (2004) and Srinivasulu and Jain, (2006) conducted studies comparing the performance of different learning algorithms in R-R modeling. Hsu et al., (1995) and Dawson and Wilby, (1998) concluded that using wetter period of R-R time series in training set can improve model performance. Minns and Hall, (1996) raised the importance of standardization of input data in improving ANN training. Later, Furundzic, (1998) showed that a decrease in input data dimension can decrease the complexity of ANNs and hence will improve the training performance. Shamseldin et al. (2002) investigated the effect of the choices of transfer function type for neurons in the hidden and output layer on the performance of ANN. Castellano-Mendez et al., (2004) showed that including high peak discharges in training set can improve the peak estimation ability of the ANN model. Agrawal and Singh, (2004) showed that using data with relatively high variability and uncertainty in the training data set can improve model performance.

After a decade of employing ANNs as a modelling tool in various hydrologic problems including R-R modelling, several issues have been raised in the literature including by Gaume and Gosset (2003) and Govindaraju (2000b) which criticize the use of ANN model. The major issues are as follow:

1- Selecting an optimal training data set is a major issue in applying ANNs in hydrologic modelling. Employing the most informative data set as training data set is desirable since it can affect the model performance seriously. For example, considering seasonal trends in rainfall-runoff modelling data (Jain and Kumar, 2007) could be one of the attempts to achieve such optimal training data sets.

2- It is believed that training ANNs is not an adaptive process. In other words, a model which is trained with older data probably cannot be used in future; hence, it needs to be retrained periodically to get an updated in response to changes in the R-R characteristics.

3- ANN is known as a black-box model which suffers from low interpretability. It is believed that ANNs can not reveal any physics of the problem. In other words, ANN is not capable to explain, in a human comprehensible form, how the model gets to its final decision (Mitra and Hayashi, 2000). Some efforts (Jain et al., 2004; Sudheer, 2005; Lee et al., 2008) have been made to extract some physical understanding of the problem by investigating the internal behaviour of the network.

Fuzzy set theory has been used in many fields of application, such as pattern recognition, data analysis, system control, etc. since its development by Zadeh (1965). Recently the research focus on neural networks has shifted from a black box approach to a semantic-based fuzzy neural architecture (Ang and Quek, 2005). Neuro-Fuzzy System (NFS) can be defined as an integration of the fuzzy systems and neural networks (Lin and Lee, 1996). In other words, fuzzy systems use the learning algorithm derived from or inspired by neural network theory to determine its parameters (fuzzy sets and fuzzy rules). Fuzzy systems have the significant advantage of reduced training time in comparison with ANNs due to its smaller dimension and being initialized with parameters relating to the problem domain (Maguire et al., 1998). Moreover, one of the advantages of NFS over ANN is presenting human-comprehensible knowledge as rules which can be simply verified, transmitted, and expanded by human experts (Mitra and Hayashi, 2000). Recently, neuro-fuzzy systems have become popular, taking advantage of the low-

level learning ability of neural networks and the high-level reasoning ability of fuzzy systems (Cho et al., 2009).

Fuzzy models that assume local model presentations with local function dynamics at the consequent or rule-layer of the models are known as Takagi-Sugeno-Kang or TSK (Takagi and Sugeno, 1985) models. In this model, the output is calculated by performing fuzzy interpolations of simpler functional models in the neighboring fuzzy partitions. The ability to accurately model a system, either globally or locally is the significant advantage of TSK-models (Quah and Quek, 2006). Specifically, the accurate global learning ability of TSK-models has motivated various practical applications including nonlinear system estimation (Yen and Langari, 1999), stock market prediction (Chang and Liu, 2008), image processing (Mariño et al., 2005), speech filtering (Thevaril and Kwan, 2005), and traffic flow modelling (Purnomo et al., 2009). One of the main criteria to categorize existing TSK-models is locality of learning. This criterion depends on the model's learning objective function, which is a minimization of the global or the local learning errors (Quah and Quek, 2006). Adaptive Network-based Fuzzy Inference Systems (ANFIS) is an example of such a TSK-model in which the global parameter tuning has been considered by means of minimizing the global error of the model (Jang, 1993). ANFIS has been found to be an appropriate tool in nonlinear mapping problems such as R-R modeling.

In general, the concept of fuzzy theory and application of neuro-fuzzy systems especially ANFIS has found expression in many papers relating to R-R modeling since the early 2000s. Mahabir et al. (2003) developed a fuzzy expert system to forecast the seasonal runoff for two catchments located in Alberta, Canada. The domain between 25% quartile to 75% quartile of the fall and spring precipitation as well as natural runoff were considered as 'average' data while values lower than 25% quartile and higher than 75% quartile were assigned as 'low' and 'high' data, respectively. After developing membership functions for each input data based on the low, average, and high values, a rule base for the fuzzy logic model was developed using three years historical data. The results of this fuzzy expert system were compared with regression models. It was concluded that the fuzzy model performed well in all ranges of data (low, medium, and high) while regression

models were just able to model high value data for both catchments. Nayak et al. (2004) applied ANFIS to model the daily discharge of the Baitarani River, India, and compared their model results with the results from the ANN and Auto Regressive Moving Average (ARMA) models. They developed 6 different models varying the number of antecedent discharge from 1 to 6 in the input vectors, in order to find the optimum number of inputs. The ANFIS model with 2 inputs was found to be the best compared to the other 5 models. The best performing ANFIS model was reported to out-perform ARMA but was similar in performance with an ANN model with 2 neurons in the hidden layer; however, ANFIS was much better in peak estimation compared to ANN. Nayak et al. (2005b) applied a neuro-fuzzy model for short-term flood forecasting based on hourly rainfall and runoff data of Kolar basin, India. Three rainfall lags at $R(t-7)$, $R(t-8)$, and $R(t-9)$ and one runoff lag at $Q(t-1)$ were selected as the inputs. Model results from ANFIS, Fuzzy Inference System (FIS), and ANN models were compared and it was found that ANFIS captured the non-linearity in the rainfall-runoff process better than the other models. Vernieuwe et al. (2005) developed three different Takagi-Sugeno type fuzzy inference system to model rainfall-runoff process in Zwalm catchment, in Belgium. Grid partitioning and fuzzy clustering with two different algorithms namely subtractive clustering and Gustafson-Kessel algorithm were used for identifying the antecedent membership functions and the least square error method was used to compute the consequent parameters. The objective of all models was to predict $Q(t+1)$ by using $R(t)$ and $Q(t)$ as input variables. The results of the three Takagi-Sugeno models were compared with results from linear regression. It was concluded that the proposed models outperformed the linear regression model. Among the three Takagi-Sugeno models, the one based on Gustafson-Kessel algorithm was found to outperform other Takagi-Sugeno models and the authors also reported an overestimation of low flows by all proposed models. Chen et al., (2006) compared the ability of ANFIS and Back-Propagation Neural Network (BPNN) for flood forecasting in Choshui River catchment, Taiwan. ANFIS was found to be superior to BPNN and the inclusion of rainfall antecedents in input signals was found to be effective in improving the accuracy of the model. Jacquin and Shamseldin (2006) employed a Takagi-Sugeno fuzzy inference system for

rainfall-runoff modeling in 6 different catchments. The authors developed two types of models one accounted for catchment wetness in the R-R process while the second one employed seasonality as a source of nonlinearity in R-R transformation. The results of this NFS model were compared with those of the Simple Linear Model, the Linear Perturbation Model and the Nearest Neighbour Linear Perturbation Model. It was revealed that NFS model can be a suitable alternative to the other modelling tools. Moreover, it was found that increasing the complexity of the model structure does not necessarily improve fuzzy model performance while the type of input variables has more significant impact on the model performance. Aqil et al. (2007) compared ANFIS for rainfall-runoff modeling in Cilalawi river basin, in west Java province Indonesia, with two different neural networks namely Levenberg–Marquardt feed forward neural network and Bayesian regularization feed forward neural network. The authors investigated the effect of data standardization in model performance and reported that standardization improved the performance. The comparative study between the three models of this study revealed that there is no significant difference between them; however, ANFIS showed slightly better results followed by Levenberg-Marquardt and Bayesian regularization feed forward neural network. Remesan et al. (2008) compared the capabilities of two artificial intelligence tools namely neural network auto regressive with exogenous input (NNARX) and ANFIS for R-R modelling in Brue catchment in United Kingdom. It was found that the results of ANFIS and NNARX are comparable and both models gave accurate and reliable forecasting results. Firat and Gungor (2008) conducted a comparative analysis between ANFIS, Feed Forward Neural Networks (FFNN), Generalized Regression Neural Network (GRNN), and Auto Regressive (AR) models for river daily flow forecasting in two catchments namely Seyhan and Cine located in south and west of Turkey, respectively. Runoff antecedents were the only input used in Seyhan catchment while in Cine catchment both rainfall and runoff inputs were employed. The results revealed that in Seyhan catchment, ANFIS was the best model followed by GRNN, AR, and FFNN models. In Cine catchment ANFIS was also the best model followed by ARX, GRNN, and FFNN models. Mukerji et al. (2009) made a comparative study for flood forecasting by ANFIS, ANN, and the Adaptive Neuro-

GA (Genetic Algorithm) Integrated System (ANGIS) models. ANGIS is an ANN model where its weights are optimized by genetic algorithm. The data of 6 rainfall gauge stations and one discharge station of Ajay River catchment in Jharkhand, India, for the period 1993-2000 were used in this study. From the available R-R hourly time series, twenty R-R events were isolated to be used. Rainfall and discharge antecedents were fed to the models to estimate discharge forecasts at 3, 6, and 9 hours ahead. Comparisons showed that ANGIS was the best model followed by ANFIS and ANN. Lohani et al. (2011) compared ANN, Takagi-Sugeno fuzzy model, and linear transformation function (LTF) for daily rainfall-runoff modeling in Narmada basin, Central India. The authors investigated model performance based on different input variables. Eleven different input combinations of 4 different inputs were used: (i) rainfall only; (ii) rainfall and antecedent moisture content; (iii) rainfall and runoff; and (iv) rainfall, runoff, and antecedent moisture content. The inclusion of antecedent moisture content was found to improve the model performance and among the different models used, the fuzzy model gave the best results, followed by ANN and LTF.

It can be inferred from the above-mentioned studies that neuro-fuzzy systems (NFS) have good potential in rainfall-runoff modeling. In addition, comparisons made between neuro-fuzzy systems and data-driven models such as ANN and regression models show that neuro-fuzzy systems are capable of providing better predictions. However, most studies concentrate on continuous time series modeling and very few studies address event-based rainfall-runoff modeling. In events based analysis, issues such as input selection and training events selection are important and are almost never discussed in the literature. Moreover, major problems of ANN such as adaptability and interpretability still remain unsolved in the above-mentioned studies for NFS models as well. The following sections provide further review on studies relating to the afore-mentioned issues.

2.2. Input Selection

An important consideration in the use of data driven models is the choice of inputs. In contrast to physically-based models, where model inputs are well defined and are

based on physical laws such as mass and momentum conservation, there is more subjectivity when choosing the type and number of model inputs to be used in data driven models. The type of the input is problem dependent and is usually subject to the availability of data. In some studies, rainfall antecedents are used as the only input for data driven R-R models (Halff et al., 1993; Sajikumar and Thandaveswara, 1999; Chua et al., 2008). In others, both rainfall and discharge antecedents are used (Achela et al., 1998; Dawson and Wilby, 1998; Tokar and Mrkus, 2000; Riad et al., 2004; Wu et al., 2005; Nayak et al., 2005a; Srinivasulu and Jain, 2006; Aqil et al., 2007; Chua et al., 2008). Though less common, additional parameters including temperature (Furundzic, 1998; Tokar and Markus, 2000), snow melt water equivalent (Tokar and Markus, 2000), soil moisture deficit (Cheng and Noguchi, 1996), and infiltration rate (Tayfur and Singh, 2006) have been used as inputs as well.

The number of inputs to be used is another consideration. To-date several methods have been employed to find the proper choice and number of input variables. In one approach, a sequence of input time series in a time window which starts from present time to a specific time is chosen as input (Tokar and Markus, 2000; Riad et al., 2004; Firat and Gungor, 2008; Chua et al., 2008). This time window can be determined by sensitivity analysis (Tokar and Johnson, 1999; Anctil et al., 2004), using correlation analyses (Sajikumar and Thandaveswara, 1999; Sohail et al., 2008), or by considering time of concentration as a threshold to the extension of this window (Jain and Prasad Indurthy, 2003). Some studies have used a narrower time window around the most correlated input antecedent, thereby pruning unnecessary input signals (Nayak et al., 2005a, 2007). In some studies output antecedents were also considered as the choices to be selected as input beside other input variables. In such case, a combination of cross-correlation, auto-correlation, and partial auto-correlation analyses have been used to find the best input combination (Dawson et al., 2002; Sudheer et al., 2002; Jain et al., 2004; Srinivasulu and Jain, 2006; Aqil et al., 2007; Nayak et al., 2007; Mutulu et al., 2008). Maier and Dandy (1997) used Haugh and Box (1977) method for input selection in which a pre-whitening process by fitting a univariate ARMA type model in each time series and calculating the model residuals was carried out. Cross-correlation analysis between the output and

inputs residuals is then used to find the best inputs. de Vos and Rientjes (2007) employed correlation coefficient and nonlinear average mutual information (AMI) between output and the possible inputs to find the best input combination in a rainfall-runoff modeling by ANN. Bowden et al. (2005) and He et al. (2011) used partial mutual information (PMI) as a tool for finding the best input combination. In the catchments with long memory where many rainfall inputs may be needed, a pre-processing of rainfall inputs by linear transformation function can be achieved. In this method a rainfall index will be defined as a weighted sum of m rainfall inputs in a linear format. The weights of such linear function can be viewed as a discrete pulse response ordinate which in turn can be obtained in parametric form by using a two-parameter Gama distribution model proposed by Nash, (1957). This methodology has been used in several studies including Shamseldin (1997), Jacquine and Shamseldin (2006), and Shamseldin (2010). Remesan et al. (2008) and Noori et al. (2011) employed Gama Test to estimate the best mean-squared error that can be achieved by a model for a given selection of inputs. Regression models can be also used to select the input variables for a data-driven model such as ANN and NFS. Hsu et al. (1995) used linear least square simplex (LLSSIM) as a tool for input selection. Stepwise regression was employed by Furundzic (1998) and Dawson and Wilby (1999) for finding the proper inputs. Auto regressive moving average (ARMA) model was successfully used for input selection by Dawson and Wilby (1999; 2001). Noori et al. (2011) used principle component regression (PCR) method for input identification. Moreover, neuro-computational techniques such as univariate (or bivariate) ANN (Maier and Dandy, 1997) and self organization map (SOM) (Bowden et al., 2005) were also used to select the inputs.

2.3. Training Event Selection in an Event-based R-R Modeling

Rainfall-runoff modeling can be carried out either on a continuous basis or event-based. Most studies in the literature have focused on continuous modeling; however, event-based analysis is also important, especially for small urban catchments, for realistic short term predictions, and also for flood forecasting applications (Tayfur and Singh, 2006). A brief review of the works where the ANN and ANFIS models have been employed in event-based analysis, with an emphasis

on the methodologies adopted in selection of training events is summarized here. Halff et al. (1993) proposed a three layered feed forward neural network for event-based R-R modeling in Bellevue, Washington and adopted a cross validation approach to swap the testing event among the available events to find the best set of events for training. Neural network was found to be successful in modeling the general hydrograph shape. Cheng and Noguchi, (1996) developed a Back-Propagation Neural Network (BPNN) to model R-R process in a catchment in Ziyang, Sichuan province, China. Rainfall and initial soil moisture deficit were taken as inputs and discharge as output. In this study events with the first and second highest peak discharge were chosen for training and the remaining events for testing. BPNN was found to give better results comparing to physically-based Xinanjiang model. Jain and Prasad Indurthy, (2003) employed a BPNN model for an event-based R-R modeling in Salado Creek, Bitters Road, San Antonio and compared it with two unit hydrograph and four regression models. The authors chose the events with highest peak discharge for training and the remaining for testing. The 4-layered BPNN was found to give the best performance compared to the other models. Chiang et al. (2004) developed an event-based stream flow forecasting in Lan-Yang River in Taiwan. Two static feed-forward neural networks with two different training algorithms namely back-propagation (BP) and conjugate gradient (CG) and also an ANN model with real time recurrent learning (RTRL) were compared. Peak flow rate and the sufficient length of training data set were the measures taken to split the 23 storm events of this study into training, verification, and testing sets. Based on this consideration 4 different scenarios with different training sets were defined. In general, RTRL algorithm was found to be consistently superior to other models since it performed well in all 4 scenarios. Moreover, BP and CG algorithms were found to be sensitive to training data length and also showed limited extrapolation ability in peak estimation when they are trained with events with lower peaks compared to the peaks available in testing data set. Tayfur and Singh, (2006) compared the ability of BPNN, Fuzzy Logic (FL) model, and a Kinematic Wave Approximation (KWA) model. Events with different values of rainfall intensities and peak flow rates were chosen for training the models. BPNN and FL models were found to be comparable to KWA; however, they required

longer historical data comparing to KWA. Sohail et al., (2008) analyzed 14 years of data for two small sub-basins in Tono, central Japan and developed a BPNN for event-based R-R modeling. Events within the first 9 (1990-2000) years were chosen as training events while events from 1999 to 2000 and 2001 to 2003 were chosen as verification and testing sets, respectively. In general, the BPNN model was found to be superior compared to an Auto-Regressive Moving Average (ARMA) model. Chua et al., (2008) developed an ANN model and compared it with kinematic wave model (KWM) to simulate event-based R-R process on an experimental overland plane. Both peak discharge (low, medium, and large) and hydrograph shape (single, double, and multiple peak events) were considered in the selection of training events. It was concluded that ANN models which contain both rainfall and discharge in their inputs outperformed the KWM while ANN models that used rainfall alone showed poorer performance.

2.4. Adaptability

Shim et al. (2002) employed ANN for real-time flood forecasting as a part of a spatial decision support system (SDSS) for flood control in Han River basin, in South Korea. SDSS is a combination of Geographic Information System, real-time meteorological and hydrological data monitoring system, and simulation and optimization sub-systems. In this study, the flood forecasting ANN model trained with inflows, reservoir water level and downstream river stages to forecast flood for a 24 hours window. The ANN model is retrained for the current flood event and ready for forecasting the next time-step, whenever new data are made available. Although this model successfully works this issue remains that ANN needs to be continuously retrained to function as a real-time flood forecasting tool. Chiang et al. (2004) developed two static feed-forward neural networks with two different training algorithms namely back-propagation (BP) and conjugate gradient (CG) for one hour ahead stream flow forecasting Lan-Yang River in Taiwan and compared them with an ANN model with real time recurrent learning (RTRL) algorithm. Unlike static feed-forward ANNs which execute fixed-weights mapping from input space to the output space, recurrent neural networks involve the initial and past state with a recurrent process. By employing a dynamic internal feed-back loop,

recurrent ANNs can store information for later use and are able to improve the learning to capture the time-varying characteristics. Rainfall and runoff data from four rain and one discharge gauging stations were used in this study. Four different scenarios based on the length of training data set and maximum peak discharge in training data were defined. Since conjugate gradient learning algorithm was found to be superior to common back-propagation further comparison was carried out between CG and RTRL models. Their results showed that CG can only be comparable with RTRL when the training data set is large enough and the peak discharge in the training set is close to that used in the testing data set. RTRL was generally superior and was found to be insensitive to the length of training data set used. Although RTRL showed significant improvement over common static neural networks, it is not adaptable, i.e. learning is stopped once training is completed. . Luna et al. (2007) proposed a recursive learning algorithm based on Expectation Maximization (EM) algorithm to develop a Takagi-Sugeno fuzzy system for monthly stream flow forecasting for the Furnas hydroelectric plant, located in Southeast of Brazil. The learning algorithm includes two steps, namely initialization and online adaptation and the data set is split into two parts to be used in these two phases. The initial rule base is started by two rules that are adjusted by EM algorithm followed by constructive offline learning which increases the number of rules. During the initialization phase some new rules may be added to the rule-base while some may be deleted. At the end of the initialization phase the final rule-base will be used in the online adaptive learning stage. In this phase there is no need to retrain the model globally since the existing model is just retrained for the new incoming pattern. The results obtained by the proposed model were compared with that obtained by ANFIS, constructive fuzzy system model (C-FSM), and an autoregressive model. It was found that the performance of all models were almost identical. However, the authors state that their proposed model is faster and has a more compact structure compared to other models. Hong and White, (2009) introduced a Dynamic Neuro-Fuzzy Local Modeling System (DNFLMS) which is an on-line Takagi-Sugeno type NFS and employed it in forecasting the flow in Waikoropupu Springs, located in Taka Valley, South Island, New Zealand. The authors employed 17 inputs including three rainfall antecedents, and five flow

antecedents from Waingaro River, 5 flow antecedents from Takaka River, and 4 flow antecedents from Cobb hydropower station, and predicted the one-day ahead flow in Waikoropupu Springs. The authors reported that DNFLMS outperformed the traditional back-propagation multi-layer perceptron (BP-MLP) neural network and required shorter training time compared to ANFIS. Despite the appeal of on-line learning proposed in this study, learning is stopped at the end of training phase and the fixed rule-base obtained during training will be used in testing phase. Moreover, modest improvement to model performance was achieved by DNFLMS when compared to ANFIS in which the offline learning algorithm is used. Hong (2012) proposed a multi-layer perceptron (MLP) with the improved sequential extended Kalman filtering learning algorithm with updating of noise covariance (EKFQ). The proposed MLP-EKFQ model was used for a real-time warning system which predicts temperature (2 hr ahead) affected by the discharge of cooling water in 1 km downstream of a thermal power station in Waikato River, in the North Island of New Zealand. In this model, the weights of neural network evolve sequentially in time and the noise matrices of EKF algorithm will be computed automatically. Through a correlation analysis, among 19 available variables, 13 variables which are highly correlated with temperature were selected. The results show that MLP-EKFQ is comparable with MLP-BP and ANFIS. Unlike MLP and ANFIS models, it is not necessary to split the data set into training and testing sets for the proposed MLP-EKFQ, since a sequential learning algorithm is used. This allows the model to adapt with the arrival of each new data point. MLP-EKFQ showed superior results to ANFIS and MLP in terms of both accuracy and computational costs. Although this study proposed a successful real-time adaptive neural network, the temporal changes that may be captured by such adaptive model will be remained masked due to the black-box nature of the model. In other words, it is not possible (or very difficult) to extract any physical understanding of the problem.

2.5. Interpretability

Wilby et al. (2003) examined the internal behavior of the neural network in a R-R modeling problem to find a physical interpretation for the architectural features of

the ANN. In this study, seven input variables of a calibrated conceptual model that were selected based on correlation analysis were fed as input to an ANN model. After choosing a parsimonious structure in terms of hidden neurons, the contribution of each hidden neuron were investigated based on qualitative and quantitative analyses of the outputs. The authors concluded that different hidden neurons are linked with quick flow and base flow components and also a threshold state in the soil moisture accounting. Sudheer and Jain (2004) investigated the internal behavior of ANN in a river flow modeling study. Investigating the output of hidden neurons revealed that these values map the high, low, and medium magnitude portions of transfer function. The authors also reported an overlap between these output sub-domains resulting from the hidden neurons. Further analyses revealed positive correlation between the outputs of several of the hidden neurons with the observed flow while the output of other hidden neurons showed negative correlation with observed flow. Jain et al. (2004) conducted an investigation to identify the physical processes inherent in neural networks trained for a rainfall-runoff modeling problem. A correlation analysis between the outputs of four hidden neurons and the five inputs (three rainfall antecedents and two discharge antecedents) revealed that three hidden neurons were correlated with discharge antecedents while one hidden neuron was correlated with rainfall antecedents. Comparisons of the outputs from the four hidden neurons with a calibrated conceptual R-R model showed that the first neuron was correlated with base flow while second and fourth neurons were correlated with surface flow and the third neuron the actual incremental infiltration. Sudheer (2005) conducted a sensitivity analysis to identify the effect of each input variable on the output of a discharge forecasting neural network for Narmada catchment, in India. The ANN model had six discharge antecedents of $Q(t-1)$ to $Q(t-6)$ (hourly basis) to predict discharge at present time $Q(t)$. For each individual input variable different error percentages (10%, 20%, 30%, 40%, and 50%) were imposed to investigate the influence of each input on the final output of the model. The results revealed that $Q(t-1)$ has significant effect on all ranges of the computed flow while $Q(t-2)$ didn't show any effect on peak flow estimation. It was also found that $Q(t-2)$ is least effective in the simulation of the rising limb compared to lean flow portions.

Similarly, $Q(t-3)$ and $Q(t-4)$ were also found to be more effective in lean flow portions of the hydrograph; however, $Q(t-4)$ was found to have a key role in prediction of low flows. Finally, $Q(t-5)$ and $Q(t-6)$ were found to be ineffective in simulating any range of the flow and it was concluded that they are redundant inputs which just increase the complexity of the model. Nayak (2010) studied the internal behavior of ANFIS during discharge forecasting for Anadapur gauging station located in Baitarani basin, India. Six different ANFIS models each including one discharge antecedent ($Q(t-1)$ to $Q(t-6)$) were trained and tested to find the effect of each individual input in predicting output, $Q(t)$. A further 10 ANFIS models with different input combinations were also developed for comparison. Models which contained the inputs $Q(t-1)$, $Q(t-2)$, and $Q(t-3)$ were found to be the best for predicting $Q(t)$. With the aid of distance-based clustering method, the input-output domain was categorized to low, medium, and high values. Investigation of the effect of change in membership grades for two sample rules representing low and high flows revealed that $Q(t-3)$ has less influence on $Q(t)$. Moreover, it was revealed that as the lag time of input variable increase the model performance deteriorated. Zhang et al. (2012) developed a Takagi-Sugeno fuzzy inference system for river flood forecasting at Guchengzi Station at Nuomin River, Dedu Station at Namor River and Ayanqian Station, upstream of Nenjiang River. In this study, discharge data was divided to low, medium, and high fuzzy subsets. After constructing the fuzzy rule-base, association rule analysis was employed to simplify the rule-base. This method finds the forecasting knowledge from historical data and tries to simplify the rule-base by pruning rules that are ineffective in increasing the forecast precision. The results revealed that the pruned rule-base shows clearer physical meaning of the model and avoids having too many unused rules.

2.6. Discussion

As it can be inferred from the studies reviewed in Section 2.2, although various methods of input selection for data-driven models have been used in R-R modeling, no specific study has been done to evaluate the selection of inputs in an event-based rainfall-runoff modeling. This study is in need due to the sensitivity of data-driven models to their inputs and also since one event is unique from another.

The review of studies mentioned in Section 2.3 reveals that the influence of physical characteristics of hydrographs in training event selection for event-based rainfall-runoff modeling has not been addressed sufficiently in the literature. For example, despite its potentially importance role on the performance of data driven models, the lag time has never been considered. The influence of hydrograph shape and lag time are masked out when the continuous modeling approach is adopted, since these factors are integrated when many events are combined, as is the case with modeling on a continuous basis.

The review of studies carried out on adaptability in ANN and NFS models (Section 2.4) reveals that these efforts suffer from at least one of the following drawbacks: (i) having a learning process limited to training phase only; therefore, model's ability to deal with any temporal changes will be limited to the knowledge it has learned during training. (ii) being a black-box model where little (or no) information of the physics can be extracted.

As can be seen in the afore-mentioned studies highlighted in Section 2.5, efforts have been made to reveal the internal behavior of neural network and link them with some physical understanding of the problem. However, in most cases a physically-based model (such as a conceptual R-R model) is needed to conduct such an investigation. Moreover, an ANN solution with good generalization does not necessarily imply involvement of hidden units with distinct meaning. Hence, any individual unit cannot essentially be associated with a single concept or feature of the problem domain (Mitra and Hayashi, 2000). Therefore, the reliability and justifiability of such studies can be open to question. In contrast to ANNs, knowledge can be directly extracted in terms of IF-THEN rules from different types of NFS. Therefore, NFS models could be generally considered as more interpretable models compared to ANNs. However, the degree of interpretability in different NFS models is itself dependent on the type of fuzzy inference system used. Compared with Mamdani-type (Mamdani and Assilian, 1975) NFS where the consequent part of the rules are linguistic (such as low and high), the consequent of rules in Takagi-Sugeno-type (Takagi and sugeno, 1985) NFS is a linear function. Therefore the Takagi-Sugeno fuzzy reasoning is not as transparent compared to the Mamdani-type (Monjezi and Rezai, 2011) NFS where an implementation procedure

is needed to transform the precise values resulted by the linear function to a linguistic term.

CHAPTER 3 METHODOLOGY

3.1 Introduction

This chapter presents some of the more fundamental concepts on fuzzy logic and the more commonly known neuro-fuzzy systems. Applications of more advanced neuro-fuzzy systems including dynamic evolving neural fuzzy inference systems that employ online learning and Pseudo outer product fuzzy neural networks which employ similarity-based fuzzy neural networks are elaborated in Chapters 6 and 7, respectively. The fundamental theories of physically-based rainfall-runoff models, the results of which are adopted from other studies for comparison, and also statistical models used as bench mark are briefly presented in Section 3.3 and 3.4, respectively. General information on study sites and data used, data standardization, data smoothing, verification and input selection methods are provided in Sections 3.5 to 3.9, respectively.

3.2 Fuzzy Logic Systems

3.2.1 Introduction to fuzzy sets and fuzzy logic

Fuzzy logic which calibrates vagueness is based on the concept of fuzzy sets. Fuzzy sets which were first developed by Zadeh (1965) are defined as sets with no crisp or clear boundary. Fuzzy set theory resembles human reasoning in the employment of approximate information and uncertainties to make decisions. Unlike two-valued Boolean logic, fuzzy logic is multi-valued and it deals with degrees of membership and degrees of truth. Fuzzy logic uses any logical value from the set of real numbers between 0 (completely false) and 1 (completely true) which is known as its membership value and the function that represents such values is called a membership function (Karray and De Silva, 2004). For example, in the universe of discourse, one membership function for “weather temperature” fuzzy set could be “high”. The membership function of the fuzzy set “high” could be shown in different ways. Figure 3.1 depicts two typical membership functions one as a smooth curve and the other as a piecewise linear function, namely a trapezoidal function. Other forms of membership functions include triangular, trapezoidal, Gaussian and sigmoidal. The membership function chosen is based on specific

applications. Fuzzy logic has its own logical operations such as “complement”, “union” and “intersection”. For each of these operations, several definitions may be found in literature, but those suggested by Zadeh (1965) are widely used (Karray and De Silva, 2004).

Figure 3.1: Typical membership functions for a fuzzy set. (Left: a smooth curve function. Right: a piecewise linear function)

Following Zadeh (1965), if A is defined as a fuzzy set in a universe X with membership function $\mu_A(x)$, its complement A' is also a fuzzy set with membership function $\mu_{A'}(x)$ which is defined as:

$$\mu_{A'}(x) = 1 - \mu_A(x) \quad \forall x \in X \quad (3.1)$$

This operation which is known as the negation (NOT) is shown schematically in terms of a membership function of fuzzy set in Fig. 3.2.

Figure 3.2: Complement (NOT) of a fuzzy set.

The membership function of the union (disjunction, *OR*) of two fuzzy sets A and B ($A \cup B$) in the same universe of discourse X is given by:

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad \forall x \in X \quad (3.2)$$

The schematic representation of *OR* operation between two fuzzy sets is given in Fig. 3.3.

Figure 3.3: Union (OR) of two fuzzy sets.

Considering the same fuzzy sets, the membership function of intersection (conjunction, AND) operation between A and B ($A \cap B$) is given by:

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \quad \forall x \in X \quad (3.3)$$

Figure 3.4 illustrates the graphic representation of AND operation between two fuzzy sets.

Figure 3.4: Intersection (AND) of two fuzzy sets.

3.2.2 Fuzzy implication (If-Then rules)

Knowledge in a fuzzy-based system is usually expressed in the form of rules and these rules are used to express conditional statements that comprise logic. In the rule:

$$\text{IF } X \text{ is } x \text{ THEN } Y \text{ is } y \quad (3.4)$$

x and y are values of variables X and Y , respectively. In such conditional statements, the if-part and then-part are known as “antecedent” and “consequent”, respectively. In a fuzzy knowledge-based system, such rules become linguistic statements in which x and y are fuzzy quantities or linguistic values for linguistic variables X and Y , respectively (Zadeh, 1965). Fuzzy if-then rules are able to capture the inexact and uncertain modes of reasoning that have significant roles in human decision making ability especially in an environment of impreciseness and uncertainty (Jang, 1993). If fuzzy sets included only in the antecedent part of a rule, this rule is known as Takagi-Sugeno fuzzy if-then rule (Takagi and Sugeno, 1984). In this form of fuzzy rules, the consequent has precise (non-fuzzy) values. Hence, in Eqn. (3.4), x is a linguistic value for the linguistic variable X , and y is a precise value for the linguistic variable Y . These two forms of if-then rules are widely used in modelling and control and play an essential role in fuzzy inference systems.

3.2.3 Fuzzy inference system

Based on the basic rule of inference or *modus ponens*, the truth of the consequent of a rule can be inferred from the truth of antecedent and implication. Similarly, in fuzzy logic it can be stated that: “If the antecedent of a rule is true to some degree of membership and the implication itself is true, then it can be inferred that the consequent will be also true to the same degree” (Lin and Lee, 1996).

Fuzzy inference is a mapping process from a given input to an output in which fuzzy logic is used. An expert system which can handle this process is known as a fuzzy inference system. A fuzzy inference system or fuzzy-rule-based system is basically composed of four major components: (i) a rule base containing fuzzy if-then rules; (ii) an interface engine which performs inference operations on the rules; (iii) a fuzzification interface which transforms the crisp values into fuzzy values based on degrees of matching with linguistic variables; and (iv) a defuzzification interface which transforms the fuzzy results into a crisp output, as shown in Fig. 3.5.

Figure 3.5: Basic structure of a fuzzy inference system

Based on the method chosen for determining its output, fuzzy inference systems are categorized into linguistic and precise models. The most widely used linguistic fuzzy inference system is Mamdani's fuzzy inference system (Mamdani and Assilian, 1975). In the Mamdani fuzzy inference system, both rule antecedents and consequents are defined by fuzzy sets. The structure can be presented as:

$$\text{IF } (x \text{ is } A_1) \text{ AND } (y \text{ is } B_1) \text{ THEN } (Z_1 = C'_1) \quad (3.5)$$

$$\text{IF } (x \text{ is } A_2) \text{ AND } (y \text{ is } B_2) \text{ THEN } (Z_2 = C'_2) \quad (3.6)$$

where A_1 , A_2 and B_1 , B_2 are membership values of input variables x and y respectively; and C'_1 , C'_2 are membership values of output variables Z_1 , Z_2 , respectively. Mamdani's fuzzy inference system for two input variables, triangular membership functions, and using the Centre of Area (COA) defuzzification method is shown schematically in Fig. 3.6.

One of the most frequently used precise fuzzy models is the Takagi-Sugeno fuzzy inference system (Takagi and Sugeno, 1985). In the Takagi-Sugeno fuzzy inference system, a fuzzy rule is constituted by a weighted linear combination of crisp inputs rather than a fuzzy set. In the first-order Takagi-Sugeno fuzzy inference system, a typical rule set with two fuzzy IF-THEN rules can be denoted as:

Figure 3.6: Mamdani's fuzzy inference system.

$$\text{IF (x is } A_1 \text{) AND (y is } B_1 \text{) THEN (} f_1 = p_1x + q_1y + r_1 \text{)} \quad (3.7)$$

$$\text{IF (x is } A_2 \text{) AND (y is } B_2 \text{) THEN (} f_2 = p_2x + q_2y + r_2 \text{)} \quad (3.8)$$

where A_1, A_2 and B_1, B_2 are membership values of input variables x and y respectively; p_1, q_1, r_1 and p_2, q_2, r_2 are the parameters of the output functions f_1 and f_2 . A basic Takagi-Sugeno inference system which produces an output function f from input variables x and y by applying triangular membership functions is illustrated schematically in Fig. 3.7.

Figure 3.7: Takagi-Sugeno fuzzy inference system.

3.2.4 Neuro-fuzzy systems (NFS)

Neuro-fuzzy hybridization results in a hybrid intelligent system that synergizes artificial neural network and fuzzy logic by combining the human-inspired reasoning style of fuzzy systems with learning and connectionist structure of neural networks. According to functionality and topology, the architecture of neuro-fuzzy systems (NFS) can be divided into three major types: (1) Cooperative neuro-fuzzy systems; (2) Concurrent neuro-fuzzy systems; and (3) Hybrid neural network-based systems or hybrid neuro-fuzzy systems (Nauck et al. 1997).

(1) Cooperative Neuro-Fuzzy Systems

In these systems neural networks has an initializing role and are employed to determine the certain parameters of fuzzy rules, fuzzy sets, or weights of the fuzzy system which can be carried out either offline or online. After training, the neural network is removed and only the fuzzy system is executed. One of the major disadvantages of this type of neuro-fuzzy systems is that the structure is not totally

interpretable. Another drawback of cooperative neuro-fuzzy systems is the lack of parallelism which means the virtually independent operation of both neural network and fuzzy system. Thus, the fuzzy inference system remains unchanged during the process of learning in neural network (Karray and De Silva, 2004). Four different types of cooperative neuro-fuzzy systems have been addressed in literature based on the parameter type of the fuzzy system (Nauck et al., 1997) which are shown schematically in Figs. 3.8 to 3.11. A set of training data is used in type 1 (Fig 3.8) to determine the membership functions using a neural network. The resulted fuzzy rules then combine with a predetermined set of fuzzy rules to implement the fuzzy system.

Figure 3.8: Cooperative neuro-fuzzy system type 1 (source: Karray and De Silva, 2004).

In the second type of cooperative neuro-fuzzy systems (Fig. 3.9), a neural network is used to identify or extract fuzzy rules from training data. In this type, fuzzy system receives the predetermined information about fuzzy sets and the structure of membership.

Figure 3.9: Cooperative neuro-fuzzy system type 2 (source: Karray and De Silva, 2004).

The neural network in type 3 (Fig. 3.10) of cooperative neuro-fuzzy systems is used to update the fuzzy system structure. The adaptation is applied to the membership functions during the on-line use of the fuzzy systems. It is worth mentioning that initial fuzzy rules and membership functions must be specified in this approach.

Figure 3.10: Cooperative neuro-fuzzy system type 3 (source: Karray and De Silva, 2004).

In the fourth type of cooperative neuro-fuzzy system (Fig. 3.11), a learning approach is used to set the importance of each rule in the fuzzy system. The weighted rules are identified using a neural network.

Figure 3.11: Cooperative neuro-fuzzy system type 4 (source: Karray and De Silva, 2004).

(2) Concurrent Neuro-Fuzzy Systems

In this type of neuro-fuzzy system shown in Fig. 3.12, the neural network is employed to pre-process the inputs or post-process the outputs of the fuzzy system.

Figure 3.12: Concurrent neuro-fuzzy system (source: Karray and De Silva, 2004).

Neural network and fuzzy system in this model are not related to each other rigidly and the neural network does not initiate any changes into the parameters of the fuzzy system. The major drawback of this type of neuro-fuzzy systems is that the results are not completely interpretable.

(3) Hybrid Neuro-Fuzzy Systems

Neuro-fuzzy hybridization results in a hybrid intelligent system that synergizes the artificial neural network and fuzzy logic by combining the human-inspired reasoning style of fuzzy systems with learning and connectionist structure of neural networks. In this architecture, a layer of hidden neurons correspond to each of the task of a fuzzy inference system. Several architectures have been addressed in literature including the adopted fuzzy inference system, method of implementation, and learning algorithm (Karray and De Silva, 2004). Neuro-fuzzy systems (NFS) are generally classified into two main groups: Linguistic NFS and Precise NFS.

Linguistic NFS:

Linguistic NFS employ Mamdani-type of inference system in their structures. In this group of NFS, linguistic output data is produced from linguistic input data. This is advantageous as the consequent part of the rules is expressed in linguistic terms, which can help to provide a clearer description of the process. To date several linguistic NFS have been developed. Fuzzy adaptive learning control network (FALCON) (Lin and Lee, 1991) is one of the early linguistic NFS which has an off-line learning algorithm with a five layers structure. Neuro-fuzzy control (NEFCON) (Nauck et al., 1997) and neuro-fuzzy classification (NEFCLASS) (Nauck and Kruse, 1995) which benefit from both incrementally and decrementally

learning ability have been used for fuzzy controlling and classification, respectively. Pseudo outer product fuzzy neural network (POPFNN) which was developed by Zhou and Quek (1996) uses a self-organising algorithm to learn and initialize the membership functions of inputs and outputs. Moreover, it employs pseudo outer product (POP) learning algorithm which is a simple one-pass algorithm. Tung and Quek (2002) developed genetic self-organizing fuzzy neural network (GenSoFNN) which employs discrete incremental clustering (DIC) method to identify and prune the redundant and /or obsolete rules. Recently, more complex linguistic NFS have been developed. As an example, Singh et al. (2008) employed discrete clustering technique (DCT) to automatically generate the membership functions for fuzzy sets. This algorithm which is known as DCT-Yager is a modified version of POPFNN which is reported to be noise tolerant (Singh et al., 2008). A summary of aforementioned models is provided in Table 3.1.

Precise NFS:

Precise NFS employ Sugeno-type inference system (Takagi-Sugeno-Kang or TSK) and is able to produce numerical (non-linguistic) output from linguistic input data. The ability of accurate modelling of a system, globally or locally, is the significant advantage of these NFS models. Several different architectures have been addressed in literature for precise neuro-fuzzy systems. The most widely used precise NFS is perhaps adaptive network-based fuzzy inference system (ANFIS) which was developed by Jang (1993). ANFIS has an off-line learning algorithm and will be discussed in detail in Section 3.2.5. Kasabov and Song (2002) developed dynamic evolving neural fuzzy inference system (DENFIS) by employing an evolving clustering method (ECM). DENFIS benefits from online or local learning. Wang et al. (2004) developed modified self-organizing TSK fuzzy neural network (MS-TSK) which employs discrete incremental clustering method and has an online learning algorithm. Tang et al. (2005) used genetic algorithm for parameter tuning of the TSK model. The genetic algorithm based TSK fuzzy neural network (GA-TSKFNN) showed better performance compared to common TSK model. Fuzzy input TSK or FITSK (Quah and Quek, 2006) has an online learning algorithm. A

localized version of recursive least square algorithm is employed for parameter tuning. A summary of above-mentioned models is provided in Table 3.1.

Table 3.1: Linguistic and precise hybrid neuro-fuzzy systems.

Mamdani Type NFS (Linguistic)	Sugeno Type NFS (Precise)
FALCON (Lin and Lee, 1991)	ANFIS (Jang, 1993)
NEFCLASS (Nauck and Kruse, 1995)	DENFIS (Kasabov and Song, 2002)
POPFNN (Zhou and Quek, 1996)	MS-TSK (Wang et al., 2004)
NEFCON (Nauck et al., 1997)	GA-TSKFNN (Tang et al., 2005)
GenSoFNN (Tung and Quek, 2002)	FITSK (Quah and Quek, 2006)
DCT-Yager FNN (Singh et al., 2008)	-

Furthermore, NFS can be also categorized based on the reasoning schema adopted. Approximate reasoning is the process of concluding a possible imprecise conclusion from a group of imprecise premises (Zadeh, 1973). One of the major issues on application of approximate reasoning in rule-based expert systems is the deduction of conclusions from observations and rules in a knowledge base (Turksen and Zhong, 1990). To date, four different reasoning schemas have been mentioned in the literature: (1) Compositional Rule of Inference (CRI) which is based on the commonly used compositional rule of inference as proposed by Zadeh (1973); (2) Truth Value Restriction (TVR) inference (Zadeh, 1975a, 1975b and 1975c) which is based on the relation from the truths of the premises to the truths of the consequence of a rule; (3) Approximate Analogical Reasoning Schema (AARS) (Turksen and Zhong, 1990) which is constructed based on similarity measures; (4) Yager inference rule (Keller et al., 1992) which is based on dissimilarity measures.

3.2.5 Adaptive Network-Based Fuzzy Inference System (ANFIS)

The adaptive Network-based Fuzzy Inference System (ANFIS) by Jang (1993) is mentioned in greater detail here since it is widely used. ANFIS can be constructed as a five layer Multi Layers Perceptron (MLP) network, illustrated in Fig. 3.13:

Figure 3.13: A typical structure for an Adaptive Network-based Fuzzy Inference system (ANFIS).

Layer (1):

In this layer, the fuzzy membership values for an input variable are estimated. The output of the node i is defined as:

$$O_i^1 = \mu_{A_i}(x) \quad \text{for } i = 1, 2 \quad (3.9)$$

$$O_i^1 = \mu_{B_{i-2}}(y) \quad \text{for } i = 3, 4 \quad (3.10)$$

where x (or y) is input and μ_{A_i} (or $\mu_{B_{i-2}}$) is the fuzzy set associated with this node. This fuzzy set is characterized by the shape of a continuous and piecewise differentiable membership function. For a triangular membership function, the output O_i^1 can be computed as:

$$O_i^1 = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (3.11)$$

where a and b are the parameters that locate the feet of the triangle while c is the parameter that locates the peak of the triangle.

Layer (2):

Multiply the incoming signals from the previous layer and calculate the firing strength of the rule. The output O_i^2 of the node i can be computed as:

$$O_i^2 = w_i = \mu_{A_i}(x)\mu_{B_{i-2}}(y) \quad \text{for } i = 1, 2 \quad (3.12)$$

Layer (3):

Each node in this layer (noted by N in Fig. 3.14) computes the normalized firing strength as:

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad \text{for } i = 1, 2 \quad (3.13)$$

Layer (4):

The node i in this layer calculates the contribution of i th rule in the model output function which is defined based on the first-order Takagi-Sugeno method as:

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad \text{for } i = 1, 2 \quad (3.14)$$

Layer (5):

The single node of this layer calculates the weighted global output of the system as:

$$O^5 = f = \frac{\sum_i w_i f_i}{\sum_i w_i} = \sum_i \bar{w}_i f_i \quad (3.15)$$

ANFIS implemented in the Fuzzy Logic Toolbox (MATLAB, 2011b) was used in this study where Back-Propagation (BP) algorithm was used to modify the initially chosen membership functions and the Least Mean Square (LMS) algorithm was used to determine the coefficients of the linear output functions (Jang, 1993). Back-propagation algorithm is perhaps the most popular algorithm for training ANNs. It is a gradient descent technique that minimizes the network error function. After passing of each input pattern of the training dataset from input layer to output layer,

the error will be computed. The computed error is propagated backward through the network to each node and the connection weights are adjusted.

In Fuzzy Logic Toolbox of MATLAB, 11 pre-defined membership functions are available to be chosen for inputs (see Figure 3.14). These membership functions are: two straight line membership functions namely triangular (trimf) and trapezoidal (trapmf); two Gaussian distribution curves (gaussmf and gaussmf2); a generalised bell shape function (gbellmf); three sigmoidal functions including single sigmoidal function (sigmf), difference between two sigmoidal functions (dsigmf), and product of two sigmoidal functions (psigmf); and three polynomial-based functions including one asymmetrical curve open to the left (zmf), one asymmetrical curve open to the right (smf), and one curve with zero on both ends and a rise in the middle (pimf). In this study, the first 8 membership functions are applicable to the available data and the remaining three are discarded. For output, however, MATLAB has only two choices for membership functions which are either constant or linear functions. This limitation on output membership function choices is because ANFIS is based on Sugeno-type NFS.

(e)

(f)

(g)

(h)

(i)

(j)

(k)

Figure 3.14: Pre-defined membership functions in Fuzzy Logic MATLAB Toolbox: (a) trimf, (b) trapmf, (c) gaussmf, (d) gaussmf2, (e) gbellmf, (f) sigmf, (g) dsigmf, (h) psigmf, (i) zmf, (j) smf, and (k) pimf. Source: Fuzzy Logic MATLAB Toolbox (MATLAB 2011b).

3.2.6 Over-fitting

This usually happens when a model learn to describe random error or noise instead of the input-output association. Over-fitting usually occurs when model has too many parameters compared to the number of observed data points. Over-fitting can be detected when model shows good performance in training while its predictive performance is low. To avoid over-fitting, usually sensitivity analyses are needed to carefully choose the model parameters such as number epochs, number of membership function, etc.

3.3 Physically-Based Models

Three physically-based models were adopted to provide the predicted data from these models for comparison with the neuro-fuzzy models developed in the current study. The models used are the Kinematic Wave Model (KWM), the Stormwater Management Model (SWMM) and University of Regina Hydrologic Model (URHM) for Catchment 1, Catchment 2 and Catchment 3, respectively. In this section the theories behind these models are briefly reviewed while the results adopted from other studies adopting these techniques are presented in the following chapters.

3.3.1 Kinematic Wave Model (KWM)

This model uses kinematic wave equations and is used widely in rainfall-runoff modelling. It is a physically-based model as it utilizes physical parameters to characterize a catchment. The kinematic wave equations for flow on an overland plane are:

$$\frac{\partial y}{\partial t} + \frac{\partial Q}{\partial x} = i_n \quad (3.16)$$

$$Q = \alpha y^\beta \quad (3.17)$$

where y is the flow depth, t is time, Q is the discharge, x is distance along the plane in the direction of the flow and i_n is the net rainfall. In Eqn. (3.16), $\alpha = \sqrt{S}/n$,

$\beta = \sqrt{5/3}$ where S is the overland slope and n is the Manning's roughness coefficient of the overland surface. Equations (3.16) and (3.17) can be solved numerically (Alley et al., 1980) to find both the discharge and flow depth for different nodes at each time step. The kinematic wave model takes the net rainfall, i_n , as input, and predicts Q (or y). The only other input required is the loss rate, since the catchment parameters are specified. Chua et al. (2008) used KWM to simulate runoff in Catchment 1. Since it is not possible to determine the losses a priori, the net rainfall for simulating an event was determined based on the loss rate of the event selected for calibrating the model.

3.3.2 Storm Water Management Model (SWMM)

The Storm Water Management Model (SWMM) using the kinematic wave routing procedure (Lighthill and Whitham, 1995). In SWMM, the sub-catchments are modeled as idealized rectangular areas with the slope of the catchment perpendicular to the catchment width. Each sub-catchment is then classified into pervious and impervious sub-areas, as defined by the impervious percentage ($Imp\%$). Flow routing is performed separately for each sub-area. Horton's model was used to simulate the infiltration loss for the pervious sub-areas. Depression losses are also modeled for the pervious sub-area, but were assumed negligible for the impervious sub-areas. Other parameters for the kinematic wave method includes the characteristic width (W), the impervious percentage ($Imp\%$), the characteristic slope (S_0) which represents an average slope of the catchment, the Manning's roughness coefficient for impervious sub-areas (N_{imp}) which indicates the resistance of impervious surface against runoff, the Manning's roughness coefficient for pervious sub-areas (N_{per}) which is similar to (N_{imp}) but relates to pervious area, and depression storage for pervious sub-areas (Dp) which represents the portion of rainfall that is trapped in surface depressions. In addition, three more parameters of the Horton's infiltration model parameterizing the maximum infiltration rate (f_0) or infiltration rate at the start of rainfall, the ultimate infiltration rate (f_c), and the decay constant (k) which represents the rate of decrease in infiltration rate against time are used.

3.3.3 University of Regina Hydrologic Model (URHM)

URHM (Qin et al., 2010) is a distributed hydrologic model designed to simulate hydrologic behaviors in large watersheds. In this model the watershed is divided into sub-watersheds. The sub-watersheds are then clustered based on some key factors including meteorological, hydrological, and physiographic ones. For each cluster, one representative sub-watershed can be selected for further analysis. A series of sub-models are calibrated and verified to simulate the R-R process for each cluster. The calibration process is undertaken by means of non-linear optimization approach and the verification performance is examined based on error analysis. Finally, the identified sub-models are integrated to form the distributed model. This model relies on two assumptions: (1) model parameters are spatially heterogeneous over the watershed; (2) the rainfall-runoff relationships are homogeneous. URHM employs a variety of several inputs to get calibrated including sub-watershed area, latitude and altitude of sub-watershed, mean elevation, mean distance of each sub-watershed from nearest stream, layout of the sub-watersheds, daily precipitation, daily mean air temperature, daily dew point temperature, and daily global radiation.

3.4 Statistical Models

3.4.1 Naïve Model

Naïve model gives the previous observed for all desired forecasts. In other words, if $Q(t)$ is observed discharge time series at time t , the same time series is considered as the prediction for $Q(t+1)$, $Q(t+2)$, etc. This is the simplest model and can be used as a benchmark against other models.

3.4.2 Autoregressive Model with Exogenous Inputs (ARX)

The ARX is a widely used statistical model for time series forecasting. ARX (n_a, n_b, n_k) model output, $Q(t)$ is assumed to be related to past outputs $Q(t-i)$ and inputs (rainfall) $R(t-i)$ by the following formula:

$$Q(t) = -\sum_{i=1}^{n_a} a_i Q(t-i) + \sum_{j=1}^{n_b} b_j R(t-n_k-j+1) + e(t) \quad (3.18)$$

where n_a and n_b are the number of past outputs and inputs, respectively; n_k is the delay associated with each input; $e(t)$ is the true error term; and a_i and b_j are model parameters to be optimized. ARX has been implemented in MATLAB Identification Toolbox. In this study, ARX model fit is evaluated using three residual statistics (Hsu et al., 1995): *RMSE*, Akaike Information Criterion (*AIC*) (Akaike, 1974), and Bayesian Information Criterion (*BIC*) (Rissanen, 1978). *AIC* and *BIC* can be computed as:

$$AIC = n_{i-o} \ln(RMSE) + 2n_p \quad (3.19)$$

$$BIC = n_{i-o} \ln(RMSE) + n_p \ln(n_{i-o}) \quad (3.20)$$

where n_{i-o} is the number of input-output patterns; and n_p is the number of model parameters. It should be noted that *RMSE* values are expected to decrease as more parameters are added to the model while *AIC* and *BIC* penalize the model for having too many parameters. Therefore, it is expected to have more parsimonious models by considering *AIC* and *BIC*. ARX model is used as a bench mark to be compared with the proposed models of this study.

3.5 Study Sites and Data Used

3.5.1 Experimental catchment (Catchment 1)

The data were obtained from an outdoor experimental plot at the Nanyang Technological University of Singapore which comprised four 25 m long by 1 m wide testing sections and was equipped with rainfall and flow measurement devices. A schematic diagram of the station is shown in Fig. 3.15. Runoff data collected from the asphalt plane which has a slope of 2% and surrounded by a 1 m high concrete wall along the two longer sides and the upstream end of the section, were analysed. Weigh tanks were used to record the flow from the asphalt plane. The weigh tanks were calibrated prior to use, using an electromagnetic flow meter. Two

rain gauges, placed at 6.25 m from each end of the test section, were used to record rainfall. The rainfall data were checked for consistency and the average rainfall was used in the analysis. Rainfall and runoff data were recorded at 15-sec interval. Data for ten storm events obtained for the period between October 2002 and December 2002 were used. A summary of the 10 events used is provided in Appendix A. More details of the experimental set up can be found in Wong (2008). This catchment is denoted as Catchment 1.

Figure 3.15: Schematic diagram of the experimental station.

3.5.2 Kranji sub-catchment (Catchment 2)

The study area chosen is a sub-catchment within the Kranji Reservoir catchment, in Singapore. The total area is 5.6 km^2 , and the land use consists of 32.6% high-density residential area with the remaining parts of the catchment mainly undeveloped (Fig. 3.16). KC01 (see Fig 3.16) is the final discharge point of the catchment. The time series of discharge and rainfall measured at 5-min intervals from 16 Dec 2004 to 30 Nov 2006 were selected and a total of 66 events were constructed from the original recorded time series (Tan et al., 2009). A summary of

the events used can be found in Appendix B. This catchment is denoted as Catchment 2 in the current study.

Figure 3.16: Schematic map of Kranji sub-catchment.

3.5.3 Heshui catchment (Catchment 3)

Heshui is a tributary of the Ganjiang River which originates from the Wugong Mountains and flows from west to east through Lianhua County, Yongxin County, Ji'An County and into the Ganjiang River (Fig. 3.17). The total length of Heshui

River is 225 km and the total area of this watershed is about 2,275 km². Its topography comprises a higher western area and a lower eastern area. The climate of the studied area can be categorized as subtropical and seasonal wind climate with high heat and rainfall during summer season. The Heshui watershed has a population of approximately 1.1 million people, mostly relying on agriculture and farming for living. The average rainfall per year is about 1,500 mm. Most of the rain falls between the months of April and June, averaging about 850 mm within these months. The average yearly rate of evaporation is about 1,070 mm. The continuous daily rainfall and discharge data recorded from 1 January 1990 to 21 January 2006 at Beimen station were used in the present study. Heshui catchment is denoted as Catchment 3 in this study.

Figure 3.17: Schematic map of Heshui watershed.

3.5.4 Rönne catchment (Catchment 4)

Rönne river basin drains a large area of southern Sweden (1,900 km²) and empties into the Kattegat area of the Baltic Sea on the Swedish west coast. Around 31 % of the basin land area is fertile and highly productive agricultural land with approximately 55,000 ha under cultivation. The remainder (48% of the area) is

primarily in privately managed forest areas. A sub-catchment of Rönne named Klippan_2 (Fig. 3.18) with area of 241.3 km² was chosen for this study and the mean daily rainfall, runoff, and temperature time series from 1961 to 2003 were used. Precipitation includes rainfall, snowfall, and other processes by which water falls to the land surface, such as hail and sleet. Therefore, temperature is included in this study to take into account the snow-melt process in rainfall-runoff modelling. This catchment will be denoted as Catchment 4 in this study.

Figure 3.18: Schematic map of Klippan_2, the sub-catchment of Rönne watershed.

3.6 Data Standardization

Standardization concentrates the dispersed data to a defined interval (Minns and Hall, 1996). It is believed that it can improve the model performance. To date several methods have been used for standardizing the data. Wu et al. (2005) used common normalization based on average value $\bar{x}$ and the standard deviation. Tayfur and Singh (2006) standardized the data based on the maximum value which can be obtained by:

$$x_n = \frac{x_i}{x_{\max} + 1} \quad (3.21)$$

where x_n is the standardized data; x_i is the measured data; and $x_{\max}$ is the maximum value of the original data. Dawson and Wilby (1998) used the sum of the squares of all values (SS) to standardize the data:

$$x_n = \frac{x_i}{\sqrt{SS}} \quad (3.22)$$

One of the commonly used methods for standardizing the data is that of van Ooyen and Nienhuis (1992):

$$x_n = FMIN + \frac{(x_i - x_{\min})}{(x_{\max} - x_{\min})} \times (FMAX - FMIN) \quad (3.23)$$

where $FMIN$ and $FMAX$ are the desired minimum and maximum of the new domain, respectively; and $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the original data series, respectively;. Different values for $FMAX$ and $FMIN$ have been exercised in literature (Shamseldin et al., 1997; Rajurkar et al., 2002; Antar et al., 2006). This method was adopted in this study to standardize all data. Following Rajurkar et al. (2002), the values of 0.1 and 0.9 were chosen for $FMAX$ and $FMIN$.

3.7 Data Smoothing

Neural networks are believed to be able to handle noisy data (Zealand et al., 1999). However, employing appropriate method to pre-process the data can improve the performance of the model (Masters, 1995). Rainfall time series with short time steps (few minutes) could be noisy. One of the common methods to smoothen the data is moving average method (Janacek and Swift, 1993). In this study, the rainfall time series of Catchment 1 and 2 with 15 s and 5 min time steps, respectively were found to be noisy. Therefore, a 3-point moving average smoothing method was applied to all rainfall data of these two catchments.

3.8 Model Performance Measures

The following error statistics and goodness of fit measures were adopted in this study:

1. Root mean square error or $RMSE$ ($\text{m}^3\text{sec}^{-1}$):

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (Q_i - \hat{Q}_i)^2}{n}} \quad (3.24)$$

where in Q_i and $\hat{Q}_i$ are observed and simulated flow rates of the i^{th} observation, respectively and n is the total number of observations. The $RMSE$ accords extra importance on the outliers in the data set and is therefore biased towards errors in the simulation of high flow rates (Dawson et al., 2006).

2. Mean Absolute Error or MAE ($\text{m}^3\text{sec}^{-1}$):

$$MAE = \frac{\sum_{i=1}^n |Q_i - \hat{Q}_i|}{n} \quad (3.25)$$

MAE computes all deviations from the original data regardless of sign and is not weighted towards high flow values (Abrahart et al., 2004).

3. Nash-Sutcliffe coefficient of efficiency or CE :

$$CE = 1 - \frac{\sum_{i=1}^n (Q_i - \hat{Q}_i)^2}{\sum_{i=1}^n (Q_i - \bar{Q})^2} \quad (3.26)$$

where $\bar{Q}$ is the average observed discharge in the calibration period. This measure which was introduced by Nash and Sutcliffe (1970) has a range between 1 (perfect fit) and $-\infty$. Zero or negative CE values indicate that the mean value of the observed time series could be a better predictor than the model (Krause et al., 2005).

4. Coefficient of determination or r^2 :

$$r^2 = \left[\frac{\sum_{i=1}^n (Q_i - \bar{Q})(\hat{Q}_i - \tilde{Q})}{\sqrt{\sum_{i=1}^n (Q_i - \bar{Q})^2} \times \sqrt{\sum_{i=1}^n (\hat{Q}_i - \tilde{Q})^2}} \right]^2 \quad (3.27)$$

where $\tilde{Q}$ is the average simulated discharge. r^2 shows the degree of co-linearity between the observed and simulated time series and has a range of 0.0-1.0, with higher values indicating a higher degree of co-linearity.

5. Relative Peak Error (*RPE*):

$$RPE = \frac{|(Q_p) - (\hat{Q}_p)|}{(Q_p)} \quad (3.28)$$

where Q_p and $\hat{Q}_p$ are observed and simulated peak discharge. Values of *RPE* closer to zero indicate better estimation of peak flows. The Relative Peak Error (*RPE*) evaluates the ability of the proposed models to accurately predict peak flows.

3.9 Input Selection Methods

3.9.1 Introduction

Antecedent rainfall and discharge are the potential inputs of a rainfall-runoff model. As discussed in Section 2.2, several methods have been employed to select the proper inputs. In this study, three different methods are used. The details of these models are provided in this section.

3.9.2 Correlation Analysis

Correlation analyses can be used to determine the number of antecedent rainfall and discharge values that have significant influence on discharge prediction. Following

Sudheer et al. (2002), the auto-correlation function (ACF), partial auto-correlation function (PACF) and cross-correlation function (CCF) were used. In this method, antecedent rainfall inputs are chosen based on CCF and antecedent discharge inputs are selected based on both ACF and PACF.

3.9.3 Mutual Information-Correlation Analysis

The correlation coefficient (CC) is given by:

$$CC(x, y) = \frac{COV(x, y)}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \times \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (3.29)$$

where COV is the covariance between variables x and y ; σ_x and σ_y are the standard deviations of x and y , respectively; $\bar{x}$ and $\bar{y}$ are the average values of x and y , respectively and n is the number of data points. CC ranges within the domain $[-1, 1]$ where value of 1 and -1 indicate positive and negative perfect linear correlation while $CC = 0$ is an indication that there is no correlation between the two data series.

The mutual information, MI , between two continuous random variables X and Y can be defined as (da Costa Couto, 2009):

$$MI(x, y) = \iint f_{x,y}(x, y) \text{Log}\left(\frac{f_{x,y}(x, y)}{f_x(x)f_y(y)}\right) dx dy \quad (3.30)$$

where $f_x(x)$ and $f_y(y)$ are the probability density functions of X and Y , respectively and $f_{x,y}(x, y)$ is the joint probability function of X and Y . $MI = 0$ indicates independence between X and Y while a high value of MI indicates strong dependence between X and Y (Fraser and Swinney, 1986; Moon et al., 1995). Considering the case of Gaussian distributed variables, Eqn. (3.30) can be simplified to:

$$MI(x, y) = \frac{1}{2} \log\left(\frac{|C_{xx}| |C_{yy}|}{|C|}\right) \quad (3.31)$$

where C is the covariance matrix given by:

$$C = \begin{bmatrix} C_{xx} & C_{xy} \\ C_{yx} & C_{yy} \end{bmatrix} \quad (3.32)$$

In Eqns. (3.31) and (3.32) C_{xx} and C_{yy} are the variance of variables X and Y , respectively and C_{xy} and C_{yx} are the between-sets covariance of variables X and Y . In definition, $C_{xy} = C_{yx}^T$ where T stands for transpose operation. $|C|$ indicates the determinant of covariance matrix.

In order to find the proper inputs, CC can be calculated for rainfall with lead times ranging from $R(t)$ to $R(t-t_0)$ and the runoff $Q(t)$ where t_0 is usually extended at least up to the time of concentration. In addition, MI was calculated using all possible combinations of two rainfall lead times from $R(t)$ to $R(t-t_0)$. Those combinations of two rainfall lead times which gave small MI values and showed large CC values can be selected. In this study, the calculated CC and MI values were then normalized within the range of $[0, 1]$ and a ranking procedure was applied to determine the inputs that gave the optimal combination of low MI and high CC values. The ranking R_j is defined as:

$$R_j = \sum_{i=1}^n \hat{C}_i + \sum_{i=1}^{n-1} \sum_{k=i}^n \hat{I}_{ik} \quad (3.33)$$

where $\hat{C}_i$ is the normalized correlation coefficient between $Q(t)$ and the i^{th} rainfall lead times in j^{th} input combination; and $\hat{I}_{ik}$ is the normalized value for $1 - MI$ for i^{th} and k^{th} inputs in j^{th} input combination. Since MI gives the mutual information, $I = 1 - MI$ represents independence. Therefore, those combinations with high R_j values are the potential candidates. This procedure can be carried out for different number of inputs including 2, 3, 4, etc. In each case, the k -highest ranked input combinations were chosen as candidates for input to the model. A verification process is needed to select the best combination and number of inputs from the selected candidates.

3.9.4 Stepwise Regression

Multiple linear regression models are built from a potentially large number of predictive terms. Stepwise regression is a systematic method for adding and removing terms from a multi-linear model based on their statistical significance in a regression. The method begins with an initial model and then compares the explanatory power of incrementally larger and smaller models. At each step, the p -value of an F -statistic is computed to test models with and without a potential term. If adding a new input contributes to the model then it is retained, but all other inputs in the model are then re-tested to see if they are still contributing to the success of the model. If they no longer contribute significantly they are removed. Inputs with p -value smaller than a threshold will be selected as inputs. Stepwise Regression is implemented in MATLAB. The p -value of 0.05 is recommended (MATLAB 2011b).

CHAPTER 4 EVALUATION OF RAINFALL AND DISCHARGE INPUTS USED BY ANFIS IN RAINFALL-RUNOFF MODELING

4.1. Introduction

The review in Chapter 2 highlighted the potential for the use of NFS in rainfall-runoff modeling and forecasting. However, most of the studies mentioned have focused on the application of ANFIS with the primary objective of establishing ANFIS as a reliable tool in rainfall-runoff modeling. Although, these kinds of studies are important in their own right, other studies are needed. For example, very little is understood about the effect of the choice of inputs on ANFIS performance for event-based rainfall-runoff modeling. Secondly, most studies on rainfall-runoff modeling using ANFIS have focused on large catchments analyzing long-term rainfall and flow data, often at hourly to daily time scales, and there have been relatively fewer studies carried out for smaller catchments and shorter time scales. In addition, studies investigating the choice of inputs to be used by ANFIS for an event-based rainfall-runoff modeling applications have seldom or never been attempted. This is important, since unlike a deterministic model where model inputs are known before hand, the choice of inputs to be used in ANFIS is often not well defined. A unique feature of event-based analysis is that one event is different from another and can be distinguished by hydrograph shape, lag time, etc., and this could influence input requirements. Data from Catchment 1 was analyzed for this study. The small size of the catchment allows the ability of the ANFIS technique in modeling runoff from a small catchment to be tested. Catchment response is relatively quick for smaller catchments and the applicability of ANFIS for small catchments (in this case a rectangular plane) which approach linearity (Sohail et al., 2008) can be evaluated against linear stochastic models.

4.2. Input Candidate Selection

The basic input data for rainfall-runoff modeling is rainfall, although in most runoff forecasting studies using data-driven methods, discharge antecedents are also used.

Commonly two approaches for input rainfall selection are adopted (see Section 2.2):

(i) Selecting a sequence of rainfall time series in a time window which starts from present time to a specific time. This time window can be determined by trial and error or may be taken from the present time to the time where the rainfall antecedent is the most correlated with discharge; e.g., if $R(t-5)$ is the most correlated antecedent rainfall then $R(t)$, $R(t-1)$, ..., and $R(t-5)$ are taken as inputs. This type of input selection method is denoted as *sequential time series* in this study.

(ii) Selecting a narrower time window of rainfall around the most correlated antecedent rainfall in order to prune unnecessary input signals; e.g., if $R(t-5)$ is the most correlated antecedent rainfall, then a set of rainfall antecedents, for e.g., $R(t-4)$, $R(t-5)$, and $R(t-6)$ may be chosen as the input signals. The length of the time window in this approach is usually determined by trial and error. This type of input selection method is denoted as *pruned sequential time series* in this study.

In addition to the two approaches mentioned above, a non-sequential approach in rainfall input selection is also possible. This approach is based on choosing a minimum number of antecedent rainfall inputs which may not necessarily be sequential. The reason for choosing a minimum number of inputs is due to the fact that ANFIS is noise intolerant and the use of unnecessary input data has been found to be one of the major sources of errors in ANFIS modeling (Mitra and Hayashi, 2000). This type of input selection is denoted as *non-sequential time series* in this study. The common input selection method for choosing sequential and pruned sequential inputs (See Section 2.2) is correlation analysis which is adopted for this study; however, no specific method has been suggested in literature for selecting non-sequential inputs. In this study, three different methods namely (1) Correlation Analysis (CA), (2) Mutual Information-Correlation Analysis (MICA), and (3) Stepwise Regression (SR) are adopted to select the non-sequential rainfall inputs.

Among the 10 available events of Catcment1 (Appendix A), three events were initially selected as possible candidates for training events based on hydrograph shape: Events 2, 6, and 3 were chosen as representatives of double, multi and single peak events, respectively. In addition, care was taken in the selection of these events such that only events with moderate or high peak flow were considered in order to avoid the model being trained by an extreme low peak

event. Another three events including Events 5 (double peak), 7 (single peak), and 8 (multiple peak) were also selected as the verification events. These six events are considered as calibration data used in input selection analysis.

Cross-correlation analysis between discharge at present time $Q(t)$ and rainfall antecedents at different lead times were carried out on calibration R-R data for Catchment 1 and the results are shown in Fig. 4.1. It can be seen that the higher correlation coefficient values are obtained for lag times of $(t-4)$, $(t-5)$, and $(t-6)$. Therefore, the first two input candidates denoted as C1 and C2 employ $R(t)$ to $R(t-5)$ and $R(t)$ to $R(t-4)$, respectively. The next four candidates denoted as C3, C4, C5, and C6 use the pruned sequential rainfall time series as inputs, employing $R(t-2)R(t-3)R(t-4)$, $R(t-3)R(t-4)R(t-5)$, $R(t-4)R(t-5)R(t-6)$, and $R(t-5)R(t-6)R(t-7)$, respectively. These four models benefit from both using a group of highly correlated rainfall inputs and having less input signals compared to C1 and C2. The six above-mentioned input candidates obtained by cross-correlation analysis are summarized in Table 4.1.

Figure 4.1: Cross-correlation results between discharge at present time $Q(t)$ and rainfall at different lead times.

Table 4.1: Sequential and pruned sequential rainfall input time series selected by cross-correlation analysis for Catchment 1.

Candidate	No. of Inputs	Inputs	Remarks
C1	6	$R(t), R(t-1), \dots, R(t-5)$	Sequential time series
C2	5	$R(t), R(t-1), \dots, R(t-4)$	Sequential time series
C3	3	$R(t-2), R(t-3), R(t-4)$	Pruned sequential time series
C4	3	$R(t-3), R(t-4), R(t-5)$	Pruned sequential time series
C5	3	$R(t-4), R(t-5), R(t-6)$	Pruned sequential time series
C6	3	$R(t-5), R(t-6), R(t-7)$	Pruned sequential time series

As mentioned in Section 3.9.3, two criteria are applied in Mutual Information-Correlation Analysis (MICA) for input selection. Firstly, the inputs used should be highly correlated with the output and secondly, the inputs used should possess low mutual information. This procedure was carried out for non-sequential rainfall input selection by using calibration data for different number of rainfall inputs including 2, 3, 4, and 5. In each case, following the ranking method introduced in section 3.9.3, the three highest ranked combinations of 2, 3, 4, and 5 rainfall inputs were chosen as candidates (total 12 candidates) denoted as C7 to C18 which are summarized in Table 4.2.

The second approach for finding non-sequential inputs is stepwise regression (SR) which is discussed in Section 3.9.4. In this study, by setting the p -value = 0.05 for entry and removal seven inputs were selected which are shown in Eqn. (4.1) from the best to the worst based on the absolute values of parameter that are allocated to them by stepwise regression:

$$Q(t) = f(R(t-2), R(t-7), R(t-4), R(t-3), R(t-5), R(t-10), R(t-6)) \quad (4.1)$$

This input candidate is denoted as C19 in this study.

Table 4.2: Input candidates selected based on mutual information-correlation analysis for Catchment 1.

Candidate	No. of Inputs	Inputs
C7	2	R(t-3), R(t-6)
C8		R(t-4), R(t-7)
C9		R(t-3), R(t-7)
C10	3	R(t-2), R(t-5), R(t-8)
C11		R(t-2), R(t-5), R(t-7)
C12		R(t-2), R(t-4), R(t-7)
C13	4	R(t-1), R(t-4), R(t-6), R(t-9)
C14		R(t-2), R(t-4), R(t-6), R(t-9)
C15		R(t-1), R(t-3), R(t-6), R(t-9)
C16	5	R(t), R(t-3), R(t-5), R(t-7), R(t-10)
C17		R(t-1), R(t-3), R(t-5), R(t-7), R(t-10)
C18		R(t), R(t-2), R(t-5), R(t-7), R(t-10)

4.3. Training Event Selection

Prior to running a verification analysis to find the best models for further comparisons, it is needed to find a proper training set. As mentioned earlier in Section 4.2, among 10 events available from Catchment 1, three events (Events 2, 6, 3) were chosen as candidates to be used as training event. Different scenarios of using 1, 2, and 3 training events were considered while verification events (Events 5, 7, 8) were used to evaluate the performance of each training set. For this verification analysis all 19 candidates obtained from Section 4.2 were used while the output was discharge at present time $Q(t)$. Two triangular membership functions were chosen for each input and the epoch = 30 was chosen for this analyses. The

selected membership function type and number of epoch are just initial values for this step and later will be properly assessed and chosen in sections 4.4 and 4.5, respectively. The training set that gives the highest average performance over the 19 input candidates during simulating the verification events will be chosen as the final training set. Table 4.3 shows the average CE , r^2 , $RMSE$, MAE , and RPE values over 19 input candidates obtained during verification by models trained with seven different training sets. As can be seen, all models performs almost marginally identical to each other; however, using Event 3 as training event is found superior to other models in terms of CE , r^2 , $RMSE$, MAE , and specifically RPE values. The reason why Event 3 is superior to other two candidates could be related to hydrograph shape, peak discharge, or lag time. More discussion on this is provided in Section 5. It was thus decided to maintain only one event for training (Event 3) since no significant improvement was achieved by increasing the length of the training data set. Moreover, choosing a single event is the worst case scenario and gives the modeler an impression of model performance under the limiting conditions where only a single event is used.

Table 4.3: Verification results obtained by ANFIS models trained with different training sets in Catchment 1.

Training Set No.	Training Events	Verification Events	CE	r^2	$RMSE$ (l/s)	MAE (l/s)	RPE
1	2		0.911	0.935	0.040	0.024	0.103
2	3		0.947	0.959	0.033	0.020	0.043
3	6		0.937	0.951	0.034	0.020	0.127
4	2, 6	5, 7, 8	0.922	0.943	0.037	0.021	0.122
5	2, 3		0.916	0.950	0.038	0.022	0.125
6	6, 3		0.926	0.951	0.036	0.022	0.128
7	2, 6, 3		0.920	0.946	0.037	0.022	0.121

4.4. Choice of Membership Function (MF)

Using the Event 3 as training event and Events 5, 7, and 8 for verification, a sensitivity analysis was conducted to evaluate the influence of using different types of membership functions in model performance. The same input candidates used in Section 4.3 are used for this sensitivity analysis. Among 11 different types of MFs which are provided in fuzzy logic toolbox in MATLAB (Section 3.2.5), 8 are relevant to the available data of this study which are investigated here. The number of membership functions also changes from 2 to 5 to show the effect of increase of number of MFs as well. An epoch of 30 was assigned and fixed for all analyses. In the first stage of this sensitivity analysis, the performance of ANFIS models in simulating verification events where different types and number of MFs used are compared in terms of CE , r^2 , $RMSE$, MAE , and RPE values in Figs. 4.2(a)-e, respectively. It is worth mentioning that these values are average values over 19 input candidates used. As can be seen, using 2 MFs disregarding its type was found to give the best verification performance when compared to using 3, 4, and 5 MFs. This could be attributed to the simplicity of R-R relationship in the small experimental catchment of the study with almost linear behaviour where increasing the number of MFs not only does not improve model performance but also deteriorates it due to the unneeded complexity. Among all eight MFs, the best CE , r^2 , and $RMSE$ values were obtained by using 2 *tri*-MFs while the best MAE and RPE values were obtained by using 2 *gauss*-MFs. However, the difference between models' performance when only 2 MFs of any of the eight types are used, was not significant. It is worth mentioning that, the decreasing trend in model performance by increase in number of MFs was monotonic for *tri*, *gauss*, and *gauss2* MFs while it had fluctuations for others. For example, *pi* and *trap* MFs show better performance when using 4 MFs than using 3 or 5 MFs.

Figure 4.2: Comparison of ANFIS models' performance in terms of (a) CE , (b) r^2 , (c) $RMSE$, (d) MAE , and (e) RPE where different types and number of MFs are used.

The ANFIS model performance during training and verification was also compared in terms of CE , r^2 , $RMSE$, MAE , and RPE values for each MF while the number of membership functions was varying from 2 to 5. It was revealed that for all eight types of MFs, training performance will improves by increasing the number of membership functions monotonically while the verification performance decreases; however, as mentioned earlier the decreasing trend was monotonic for *tri*, *gauss*,

and *gauss2* MFs only. Fig. 4.3 illustrates the training versus verification performances for two MFs types namely *tri* and *gauss* while the rest were not reported here due to parsimony. Conclusively, using 2 *gauss* MFs for each input was adopted for this study.

Figure 4.3: Comparison of training versus verification performances due to the change in number of MFs used: (a)-(e) *tri* MF, and (f)-(j) *gauss* MF.

4.5. Over-fitting Issue

In MATLAB fuzzy toolbox, the default for goal error is zero. Therefore, it is needed to determine the proper number of epochs to have the model sufficiently trained while over-fitting is also avoided. For this, a sensitivity analysis was carried out by using Event 3 as training data set and Events 5, 7, and 8 as verification data set. Same input candidates were adopted and for each input 2 *gauss*-MFs were allocated. The training and verification performances of the ANFIS model while the number of epochs was varying from 1 to 100, were compared in terms of average *CE*, r^2 , *RMSE*, *MAE*, and *RPE* values over 19 input candidates and are shown in Figs 4.4(a)-(e), respectively. As can be seen, training performance improves by increasing the number of epochs gradually; however, after approximately 50 epochs it doesn't change significantly. On the other hand, verification performance also improved gradually by increasing the number of epochs till reaching epoch =19 while after that the increasing rate became smaller and finally started to deteriorate

after 43 epochs. This decrease can be considered as a trace of over-fitting where training improves by increasing the number of epochs while verification deteriorates. Therefore, it was concluded that the number of epochs should be considered more than 20 and less than 40 to be sufficiently trained while over-fitting is also avoided. Hence, 30 epochs were adopted for this study.

Figure 4.4: ANFIS performance during training and verification in terms of (a) CE , (b) r^2 , (c) $RMSE$, (d) MAE , and (e) RPE where number of epochs was varying.

4.6. Models Development

The choice between sequential, pruned sequential, and non-sequential input combinations was conducted by using Event 3 as training event and Events 5, 7, and 8 as verification events. Two *gauss*-MFs were assigned for each input and the number of epoch was fixed at 30. Table 4.4 shows the *CE*, r^2 , *RMSE*, *MAE*, and *RPE* values obtained by 19 different ANFIS models developed based on the 19 input candidates discussed in Section 4.2. These models are denoted as M1 to M19 in this study.

Table 4.4: Verification results for ANFIS models with different input candidates.

Model	Input Type	Input Selection Method	CE	r^2	$RMSE$	MAE	RPE
M1	Sequential	Correlation Analysis	0.675	0.789	0.076	0.035	0.310
M2			0.299	0.660	0.110	0.043	0.898
M3	Pruned Sequential		0.622	0.809	0.074	0.033	0.565
M4			0.733	0.858	0.063	0.029	0.459
M5			0.933	0.942	0.035	0.021	0.140
M6			0.272	0.724	0.101	0.033	1.205
M7	Non-Sequential	Mutual Information and Correlation Analysis	0.941	0.957	0.032	0.019	0.160
M8			0.941	0.956	0.033	0.020	0.086
M9			0.948	0.958	0.031	0.018	0.041
M10			0.327	0.738	0.088	0.028	0.222
M11			0.890	0.912	0.044	0.021	0.260
M12			0.858	0.906	0.049	0.0228	0.306
M13			0.768	0.824	0.0630	0.027	0.707
M14			0.578	0.709	0.086	0.032	0.546
M15			0.857	0.873	0.051	0.025	0.304
M16			0.586	0.666	0.084	0.034	0.279
M17			0.744	0.806	0.068	0.032	0.327
M18			0.829	0.851	0.054	0.027	0.454
M19		Stepwise Regression	0.664	0.754	0.078	0.037	0.320

As can be seen, among the ANFIS models with sequential input candidates (M1, and M2), M1 gives higher performance in terms of *CE*, r^2 , *RMSE*, *MAE*, and *RPE* values than M2. Among the four ANFIS models with pruned sequential inputs (M3 to M6), M5 was superior to other three models. It is also concluded that among 13

ANFIS models with non-sequential inputs (M7 to M19), using two rainfall inputs (Models M7, M8, and M9) generally gives better results comparing to other models (M10 to M19). Moreover, model M9 gave the smallest *RPE* value compared to other ANFIS models with non-sequential inputs. Therefore, three models M1, M5, and M9 were chosen as the best representatives of sequential, pruned sequential and non-sequential inputs to be used for further comparisons.

The above-mentioned rainfall inputs can be considered for a rainfall-runoff modelling problem in which $Q(t)$ is the desired output; however, for a forecasting problem using discharge antecedents as inputs can also be considered. In terms of the number of discharge inputs to be used, an auto-correlation and partial auto-correlation analysis of the discharge time series is usually recommended (Sudheer et al., 2002). The auto-correlation function (ACF) results in Fig. 4.5 show a significant correlation up to lag (t-5) and gradually decay for the next lags. The partial auto-correlation function (PACF) results and corresponding 95% confidence limits in Fig. 4.6 show significant correlation up to the (t-5) lag and then fall within the confidence band. Therefore, the maximum number of discharge antecedents to be used is 5; however, due to the small size of the catchment the rainfall-runoff relationship is expected to be close to linear behaviour. A sensitivity analysis showed that using one discharge antecedent (the most correlated antecedent $Q(t-1)$) is sufficient in this study. The use of $Q(t-1)$ as input has also been adopted by Tokar and Markus, (2000) and Chua et al. (2008).

Figure 4.5: Auto-correlation analysis for the discharge time series.

Figure 4.6: Partial Auto-correlation analysis for the discharge time series.

The final selected models for further comparisons are summarized in Table 4.5. Model 1 employs $R(t)$ to $R(t-5)$ as its inputs since the higher average correlation coefficient values with a lower deviation between the minimum and maximum are obtained for lag times of $(t-4)$, $(t-5)$, and $(t-6)$. Model 5 uses the pruned sequential rainfall time series as its inputs ($R(t-4)$, $R(t-5)$, $R(t-6)$). This model benefits from both using a group of highly correlated rainfall inputs and having less input signals compared to Model 1. Model 9 uses non-sequential rainfall inputs $R(t-3)$ and $R(t-7)$ which were chosen among 12 candidates were found by mutual information-correlation analysis. This model benefits from both using a set of optimally selected rainfall inputs by mutual information-correlation analysis and having less input signals compared to Model 1 and 5. Model 1*, 5*, and 9* which are identical to model 1, 5, and 9, respectively, except for the addition of $Q(t-1)$ as an extra input signal are considered in this study to evaluate the effect of including a discharge antecedent on model performance. Finally, Model 20 was developed by using $Q(t-1)$ as the only input signal in order to make a comparison with the extreme case where all rainfall data are excluded from the input.

One advantage in the use of the pruned sequential time series or the non-sequential time series for forecasting is that discharge can be forecasted by shifting the input rainfall window. By simultaneously shifting forward the rainfall antecedents and discharge forecasts (by the same time interval), model performance will be identical. For e.g., in Model 9 to predict $Q(t)$, $R(t-3)$ and $R(t-7)$ are used as inputs and to forecast $Q(t+1)$ the input rainfall time window is shifted one time step ahead ($R(t-2)$ and $R(t-6)$) and so on. Shifting the rainfall input forward in time

however, has a limit since the input rainfall for time $> t$ is not available. Thus, for lead times, $t_l > t + 3$, the use of a separate model is necessary where $R(t)$ was fixed as one of the rainfall inputs. Hence, to forecast $Q(t+4)$, $Q(t+5)$, $Q(t+6)$, $Q(t+7)$, and $Q(t+8)$, $R(t)$ and $R(t-4)$ were used as inputs. Fixing the inputs to $R(t)$ and $R(t-4)$ ensures that only one model is used to forecast flows for $t_l > t + 3$. The rainfall antecedents used for the estimation of $Q(t)$ to $Q(t+8)$ are provided in Table 4.6.

Table 4.5: Details of the models chosen for comparisons in Catchment 1.

Model	Inputs	Remarks
M1	$R(t), R(t-1), \dots, R(t-5)$	Sequential time series
M5	$R(t-4), R(t-5), R(t-6)$	Pruned sequential time series
M9	$R(t-3), R(t-7)$	Non-sequential time series
M1*	$R(t), R(t-1), \dots, R(t-5), Q(t-1)$	Sequential time series
M5*	$R(t-4), R(t-5), R(t-6), Q(t-1)$	Pruned sequential time series
M9*	$R(t-3), R(t-7), Q(t-1)$	Non-sequential time series
M20	$Q(t-1)$	-

Table 4.6: Rainfall inputs used for prediction of $Q(t)$ to $Q(t+8)$ in Model 9.

Antecedent rainfall	Predicted Discharge
$R(t-3), R(t-7)$	$Q(t)$
$R(t-2), R(t-6)$	$Q(t+1)$
$R(t-1), R(t-5)$	$Q(t+2)$
$R(t), R(t-4)$	$Q(t+3)$
$R(t), R(t-4)$	$Q(t+4)$
$R(t), R(t-4)$	$Q(t+5)$
$R(t), R(t-4)$	$Q(t+6)$
$R(t), R(t-4)$	$Q(t+7)$
$R(t), R(t-4)$	$Q(t+8)$

Figure 4.7(a) shows typical average CE values for discharge predictions from $Q(t)$ to $Q(t+8)$ by Model 9 where Event 3 was used as the training event and Events 5, 7, and 8 were used for verification. The rainfall inputs used for each discharge prediction are indicated below the data points in Fig. 4.7(a). As shown in Fig. 4.7(a), CE values for discharge predictions of $Q(t)$ to $Q(t+3)$ are identical since the predicted output can be obtained by shifting the time window as the input rainfall required does not exceed the present time t . Thus, discharge predictions that fall within the range (t) and $(t+3)$ are denoted as “short-term” in this study. On the other hand, the five remaining predictions of $Q(t+4)$ to $Q(t+8)$, show a decreasing trend which is to be expected since the inputs $R(t)$ and $R(t-4)$ are used for all the forecasts. Hence, discharge predictions for $t_i > t+3$ are denoted as “long-term” in this study. Based on this discussion, Model 9 thus consists of two separate sub-models, Models 9(a) for short-term and Model 9(b) for long-term discharge predictions. The CE values for Model 9* which is similar to Model 9, except that it includes $Q(t-1)$ as input, are similarly compared in Fig. 4.7(b). For $Q(t)$, $Q(t+1)$, $Q(t+2)$, and $Q(t+3)$, a similar shifting approach was used with the rainfall inputs in addition to the discharge $Q(t-1)$. The inclusion of $Q(t-1)$ gives higher CE values for short-term predictions in Model 9*(a) compared to Model 9(a). Unlike Model 9(a), CE values for $Q(t)$, $Q(t+1)$, $Q(t+2)$, and $Q(t+3)$ simulated by Model 9*(a) have a slightly decreasing trend from $Q(t)$ to $Q(t+3)$ which is due to the inclusion of $Q(t-1)$. Beyond $Q(t+3)$, CE decreases as observed for Model 9(b) but at a larger rate.

Figure 4.7: Average *CE* values for (a) Model 9 and (b) Model 9* for simulating discharge predictions from $Q(t)$ to $Q(t+8)$. Inputs used for discharge predictions are shown below the data points. (Training Event: Event 3; Verification Events: Events 5, 7, and 8).

4.7. Results and Discussions

The final 7 models of this study (see Table 4.5) were trained with Event 3 and then used to test the nine remaining events. According to the findings in Section 4.2 to 4.4, two Gaussian membership functions were used for each input and the number of epochs was fixed for 30. Initial analysis revealed that discharge predictions for $t_l > t + 8$ do not give reasonable results in terms of used statistics such as CE , r^2 , $RMSE$, and MAE ; hence, study was focused on predictions up to $t_l = t + 8$.

4.7.1. Goodness-of-fit

The gross performance of each of the models are shown by box and whisker plots of the r^2 values of the 9 events (excluding the training event) for predictions of $Q(t)$, $Q(t+1)$, $Q(t+2)$, $Q(t+4)$, and $Q(t+8)$ in Fig. 4.8(a), 4.8(b), 4.8(c), 4.8(d), and 4.8(e), respectively. The box plots indicate the 25th percentile (Q_1), median and 75th percentile (Q_3) of the error statistics. The lower and upper whiskers values indicate the values $Q_1 - 1.5(IQR)$ and $Q_3 + 1.5(IQR)$, respectively, where IQR is the inter-quartile range or $Q_3 - Q_1$. Data points greater than $Q_3 + 1.5(IQR)$ or less than $Q_1 - 1.5(IQR)$ are considered as outliers.

As can be seen, Models 1 which uses sequential rainfall time series as its inputs, generally gives the lowest r^2 values for the prediction of $Q(t)$ to $Q(t+8)$. The range between the minimum and maximum r^2 values is also large showing that this model is inconsistent in its performance. Model 5 that uses the pruned sequential rainfall time series, generally performs better than Models 1 for the prediction of $Q(t)$ to $Q(t+8)$. Comparing to Model 1, Models 5 shows low deviations in minimum and maximum r^2 values indicating that this model is able to predict all events well. Model 9 which is developed based on two separate sub-models (Models 9(a) and 9(b)) and uses non-sequential rainfall inputs gives relatively high r^2 values and smaller deviation between the minimum and maximum r^2 values compared to Models 1 and 5 for short- and long-term forecasts. For both short- and long-term forecasts, the advantage of choosing a lesser number of inputs is evident when the r^2 values for Model 9 (2 inputs) is compared with the r^2 values for Model 5 (3

inputs) and Model 1 (6 inputs). It can be noticed that Model 5 performs comparable with Model 9 for prediction of $Q(t+4)$ and $Q(t+8)$ while Model 1 is the worst since its median r^2 value is the lowest and its deviation between minimum and maximum r^2 values is the largest.

Considering Models 1*, 5*, and 9* in general, including $Q(t-1)$ as an additional input improves $Q(t)$ predictions. This is evident from the increase in r^2 for Models 1*, 5*, and 9* when compared to Models 1, 5, and 9, respectively. There is thus a definite advantage in choosing $Q(t-1)$ as input for $Q(t)$ predictions. In the extreme case, Model 20 which uses $Q(t-1)$ as its only input, performs well in predicting $Q(t)$ and also $Q(t+1)$. However, its performance gradually decreases for longer-term forecasts accompanied by larger deviations between the minimum and maximum r^2 values. This behavior is attributed to the high correlation between $Q(t-1)$ and the first two short-term predictions $Q(t)$ and $Q(t+1)$; but significantly poorer correlations when $t_i > t+1$. Beyond t (or $t+1$), r^2 for Models 1*, 5*, and 9* decreases and the performance, compared to the models that do not include $Q(t-1)$ in their inputs, is worse. This can be inferred from both the reduction of the r^2 values and the increase in deviations between the minimum and maximum r^2 values for $Q(t+4)$, and $Q(t+8)$ forecasts simulated by Models 1*, 5*, and 9*, in comparison with Models 1, 5, and 9, respectively. Among Models 1*, 5*, and 9*, Model 9* gives the best results since the median r^2 value for predicting $Q(t)$, and forecasting $Q(t+1)$, $Q(t+2)$ are above 0.9 and for $Q(t+4)$ close to 0.9 ($r^2 = 0.887$) with reasonably low deviations between the minimum and maximum r^2 values. In addition, for simulating $Q(t+8)$, Model 9* gives the highest median r^2 value compared to Models 1* and 5*; however, the deviation between the minimum and maximum r^2 values in this model is the largest.

(a) $Q(t)$

(b) $Q(t+1)$

(c) $Q(t+2)$

(d) $Q(t+4)$

Models

Models

Figure 4.8: r^2 values of the 9 events for (a) $Q(t)$, (b) $Q(t+1)$, (c) $Q(t+2)$, (d) $Q(t+4)$, and (e) $Q(t+8)$.

It can be noted that models which use rainfall alone as input are worse at short-term forecasts compared to models that include $Q(t-1)$ as input. Conversely, models that do not contain $Q(t-1)$ as input perform better at longer-term forecasts. This difference in behavior is attributed to the lag time of the discharge in response to rainfall. In using sequential or pruned sequential rainfall time series as inputs, the selected rainfall inputs are correlated most with the discharge occurring at some time interval $> t$, due to the finite time taken for discharge measured at the outlet to respond to the rainfall. Therefore, r^2 is expected to increase and then decrease as the lead time increases. However, Model 5 and 9 which benefits from using shifted rainfall time series, gives constant (high) median value up to $Q(t+2)$ and then shows a decreasing trend for $t_l > t+2$. On the other hand, Model 5* which includes $Q(t-1)$ in the input has the maximum median r^2 value at $Q(t)$, since $Q(t-1)$ and $Q(t)$ are the most correlated. However, median r^2 value decreases as lead time increases since the correlation between $Q(t)$ and discharge declines when time $> t$ (see Fig. 4.8, Model 5*). Significantly, median r^2 Value for Model 5* is initially greater than that of Model 5, but is less than that of Model 5 when $t_l \geq t+1$. Similar to Model 5*,

Model 9* which uses $Q(t-1)$ as input shows a monotonically decreasing trend from $Q(t)$ to $Q(t+8)$. Median r^2 value remains relatively high from $Q(t)$ to $Q(t+2)$. Model 9* also gives better long-term predictions compared to Model 5*, which uses pruned sequential time series (see Fig. 4.8).

To evaluate the goodness-of-fitness of the models with another measure rather than r^2 , the performances of each of the models are also compared in terms of *MAE*. Box and whisker plots of the *MAE* values of the 9 events (excluding the training event) for prediction of $Q(t)$, $Q(t+1)$, $Q(t+2)$, $Q(t+4)$, and $Q(t+8)$ in Fig. 4.9(a), 4.9(b), 4.9(c), 4.9(d), and 4.9(e), respectively. As can be seen, Models 1 which uses sequential rainfall time series as its inputs, generally gives the largest *MAE* values for the prediction of $Q(t)$ to $Q(t+8)$. The range between the minimum and maximum *MAE* values is also large showing that this model is inconsistent in its performance. Model 5 that uses the pruned sequential rainfall time series, generally performs better than Models 1 for the prediction of $Q(t)$ to $Q(t+8)$. Comparing to Model 1, Models 5 shows low deviations in minimum and maximum *MAE* values indicating that this model is able to predict all events well. Model 9 which is developed based on two separate sub-models (Models 9(a) and 9(b)) and uses non-sequential rainfall inputs gives relatively low *MAE* values and smaller deviation between the minimum and maximum *MAE* values compared to Models 1 and 5 for short- and long-term forecasts. For both short- and long-term forecasts, the advantage of choosing a lesser number of inputs is evident when the *MAE* values for Model 9 (2 inputs) is compared with the *MAE* values for Model 5 (3 inputs) and Model 1 (6 inputs). It can be also noticed that Model 5 performs comparable with Model 9 for prediction of $Q(t)$ to $Q(t+8)$.

(a) $Q(t)$

(b) $Q(t+1)$

(c) $Q(t+2)$

(d) $Q(t+4)$

Models

Models

Figure 4.9: MAE values of the 9 events for (a) $Q(t)$, (b) $Q(t+1)$, (c) $Q(t+2)$, (d) $Q(t+4)$, and (e) $Q(t+8)$.

Similar to what was seen for r^2 , including $Q(t-1)$ as an additional input improves $Q(t)$ predictions in terms of MAE . This is evident from the decrease in MAE for Models 1*, 5*, and 9* when compared to Models 1, 5, and 9, respectively. In the extreme case, Model 20 which uses $Q(t-1)$ as its only input, performs well in predicting $Q(t)$ and also $Q(t+1)$; however, its performance gradually decreases for longer-term forecasts accompanied by larger deviations between the minimum and maximum MAE values. This behavior is attributed to the high correlation between $Q(t-1)$ and the first two short-term predictions $Q(t)$ and $Q(t+1)$; but significantly poorer correlations when $t_l > t + 2$. Beyond t (or $t+1$), MAE for Models 1*, 5*, and 9* increases and the performance, compared to the models that do not include $Q(t-1)$ in their inputs, is worse. This can be inferred from both the increase of the MAE values and the increase in deviations between the minimum and maximum MAE values for $Q(t+4)$, and $Q(t+8)$ forecasts simulated by Models 1*, 5*, and 9*, in comparison with Models 1, 5, and 9, respectively. Among Models 1*, 5*, and 9*, Model 9* gives the best results since the median MAE value and the deviations between the minimum and maximum MAE values for predicting $Q(t)$, and forecasting $Q(t+1)$ to $Q(t+8)$ are smaller than Model 1* and 5*.

Similar to what was discussed based on r^2 , models which use rainfall alone as input are worse at short-term forecasts compared to models that include $Q(t-1)$ as input. Conversely, models that do not contain $Q(t-1)$ as input perform better at longer-term forecasts. Moreover, Models which uses $Q(t-1)$ as input shows a monotonically decreasing trend from $Q(t)$ to $Q(t+8)$. Among Models 1*, 5*, and 9*, median *MAE* values for Model 9* remains relatively low from $Q(t)$ to $Q(t+2)$. Model 9* also gives better long-term predictions compared to Models 1* and 5* which use sequential and pruned sequential time series, respectively (see Fig. 4.9).

4.7.2 Peak Estimation

Figures 4.10(a), 4.10(b), 4.10(c), 4.10(d), and 4.10(e) illustrate the box and whisker plots of the *RPE* values of the 9 events (excluding the training event) for predictions of $Q(t)$, $Q(t+1)$, $Q(t+2)$, $Q(t+4)$, and $Q(t+8)$, respectively. In general, Models 5 and 9 which benefit from pruned sequential and non-sequential inputs, respectively, perform better than Model 1 in peak estimation as they show smaller *RPE* values with smaller variation between minimum and maximum *RPE* values when compared to Model 1. Model 1* and 9* give lower *RPE* values with smaller deviations between the minimum and maximum *RPE* for the 9 simulated events when predicting $Q(t)$, $Q(t+1)$, and $Q(t+2)$ compared to Model 1 and 9, respectively. The better performance of Models 1* and 9* is attributed to the inclusion of $Q(t-1)$ as input since it improves the short-term predictions. However, Model 5* is an exception for this trend as including $Q(t-1)$ didn't improve the peak estimation capability of the model. In the extreme case, Model 20 which uses only $Q(t-1)$ as input, performs the best in the prediction of $Q(t)$, $Q(t+1)$, and $Q(t+2)$. Peak estimation by Models 1*, 5*, 9*, and 20 deteriorate by increasing the lead time. Models 1* and 5* especially are not suitable for long-term peak estimation since *RPE* of some events are > 1 . In general, Models 1, 5, and 9 are better than the models that contain $Q(t-1)$ for $t_l > t + 2$ predictions. Furthermore, Models 9 and 9* which use non-sequential rainfall inputs are consistent in model predictions $Q(t)$ to $Q(t+8)$ for all the events. In summary, for short-term predictions, Model 20 provides the best peak estimates followed by Models 9*, and 9. This could be

expected for Model 20 as it is excellent for short-term forecasts. For long-term forecasts however, Models 5, 9 perform better.

Figure 4.10: RPE values of the 9 events for (a) $Q(t)$, (b) $Q(t+1)$, (c) $Q(t+2)$, (d) $Q(t+4)$, and (e) $Q(t+8)$.

4.7.3 Time Shift Error

Using discharge antecedents as inputs in discharge forecasting has been found to be a source of time shift error in the simulated hydrographs. This has been reported in ANN studies by Dawson and Wilby, (1999), Jain et al. (2004), and de Vos and Rientjes, (2005) who detected time shifts in the application of ANNs in discharge forecasting. These shifts were found to be attributed to the higher weighting accorded to the discharge antecedents, relative to rainfall used in the inputs. Recently, methods to reduce time shift errors have been discussed by de Vos and Rientjes, (2005) and Abrahart et al. (2007). However, it doesn't fall within the interests of this study.

In order to evaluate the effect of different choice of inputs on time shift in discharge predictions by ANFIS, time shift errors were calculated for Models 1, 5, 9, 1*, 5*, 9*, and 20. In this analysis, the simulated discharges were shifted backward up to 10 time steps and the CE values were re-calculated after each time shift. The *time shift error* (δ) is then defined as the number of time shifts which

results in the maximum CE value. This method was also used by de Vos and Rientjes, (2005). Figure 4.11 shows a typical result for the $Q(t+8)$ forecast of Event 10 by Model 5. A time shift of zero indicates the CE value of the un-shifted, simulated time series. Thus, the maximum CE value is attained at 5 time shifts and the time shift error for the $Q(t+8)$ forecast of Event 10 by Model 5 is 75 seconds.

This analysis was repeated and the average time shift errors ($\bar{\delta}$) and their variations for all the 9 events (excluding the training event) are shown in Figs. 4.12(a) to 4.12(f) for Models 1, 5, 9, 1*, 5*, and 9*, respectively. The error bars in this figure illustrate the range between the minimum and maximum time shift errors of all the 9 tested events. The results of Model 20 are also included in the figures as dashed-lines. For Model 20 the $\bar{\delta}$ value for $Q(t)$ is 1 time lag since $Q(t-1)$ was used as input in the prediction of $Q(t)$. The $\bar{\delta}$ value for subsequent discharge predictions are thus $t_l + 1$ time lags. Also there is no variation in the time shift error for different events calculated by Model 20 since all events will have the same δ values.

Figure 4.11: Variation of CE in response to shifts of the Model 5-simulated $Q(t+8)$ discharge for Event 10.

Figure 4.12 shows that for $t_l < t+4$ predictions, Models 1, 5, and 9 have smaller $\bar{\delta}$ with lower range of the minimum and maximum δ compared to Models 1*, 5*, and 9*, respectively. The larger $\bar{\delta}$ for Models 1*, 5*, and 9* is attributed to the auto-regressive nature of the models associated with the use of $Q(t-1)$ in the inputs. Hence Model 20 which uses $Q(t-1)$ as the only input has the greatest $\bar{\delta}$, followed by Models 10, 11, and 14 which use both rainfall and $Q(t-1)$ as inputs, while Models 3, 4, and 7 which contain rainfall alone as input have the smallest $\bar{\delta}$. It is also evident that Models 1, 5, and 9 have small deviations between the minimum and maximum δ values. Among Models 1, 5, and 9, Model 9 gives the lowest $\bar{\delta}$ value. Hence, the use of rainfall as the sole input results in smaller time shift errors and the time shift errors are smallest when the non-sequential rainfall time series are used as input. For $t_l > t+4$ discharge forecasts, the $\bar{\delta}$ value for Models 1*, 5*, and 9* are smaller than Models 1, 5, and 9, respectively. However it can be noticed from Fig. 4.12 that variation between the minimum and maximum δ values vary over a wide range. Hence, it is questionable if any of the models is robust enough to provide $Q(t+8)$ prediction with small $\bar{\delta}$ values consistently. Model 9, however is an exception, revealing a $\bar{\delta}$ value of 47 seconds (roughly 3 time shifts) and small minimum-maximum range of δ for the 9 events modeled. The $t_l = t+4$ discharge forecasts act like a transition beyond which the models (with the exception of Model 9) are not able to make predictions with consistently small $\bar{\delta}$ values.

Figure 4.12: $\bar{\delta}$ values in discharge predictions for all nine events simulated by models (a) 1, (b) 5, (c) 9, (d) 1*, (e) 5*, and (f) 9*.

4.8 Models Comparison

The preceding analysis compared and discussed the performances of the 7 models in terms of goodness-of-fit, peak estimation, and time shift error. A ranking procedure is employed to unify the relative performance of each model based on the average CE , r^2 , $RMSE$, MAE , RPE , and δ values of 9 testing events. For each statistic, the average values resulted by the 7 models are sorted and ranked from 1 (the best) to 7 (the worst). Then the sum of this ranking numbers is calculated for each model for simulating $Q(t)$, $Q(t+1)$, $Q(t+2)$, $Q(t+4)$, and $Q(t+8)$, separately. The model with small ranking summation is better than a model with large ranking summation. Based on the proposed ranking method, a model that can be ranked first for all predictions will achieve score 7 while the model that can be the worst for all predictions will be scored as 49. Table 4.7 shows the ranking of the models for the prediction of $Q(t)$, $Q(t+1)$, $Q(t+2)$, $Q(t+4)$, and $Q(t+8)$. It can be observed from Table 4.7 that no single model is ranked first for all the discharge predictions.

For $Q(t)$, $Q(t+1)$, and $Q(t+2)$, Model 9* is the best due to the high correlation between $Q(t)$ and $Q(t-1)$. Significantly, for $Q(t+4)$ and $Q(t+8)$ Model 9 is the best. The change in the models ranked first (M 9*, M 9*, M 9*, M 9, M 9) as predictions proceed from $Q(t)$ to $Q(t+8)$ reflects the relative significance of including $Q(t-1)$ as an input and shows that models which contain discharge antecedent in the input are better in short-term predictions ($Q(t)$, $Q(t+1)$, and $Q(t+2)$) while models that use rainfall only perform better for longer time step predictions ($Q(t+4)$ and $Q(t+8)$). It is observed that, Models 9 and 9* which use a non-sequential rainfall time series are ranked higher and perform well for both short and long term predictions. It can be concluded that models which use non-sequential rainfall inputs (Models 9 and 9*) are suitable models for the entire range of discharge predictions. Among models using the pruned sequential time series (Models 5 and 5*), Models 5* ranked higher in all predictions compared to Model 5 except for $Q(t+8)$ in which Model 5 performed better than Model 5*. Models using sequential rainfall time series (Models 1 and 1*) were found to be unreliable since these models are consistently ranked in the lower positions.

Table 4.7: Ranking of the 7 models of Catchment 1 based on the proposed ranking procedure.

Rank	Q(t)	Q(t+1)	Q(t+2)	Q(t+4)	Q(t+8)
1	M 9*	M 9*	M 9*	M 9	M 9
2	M 20	M 9	M 9	M 9*	M 9*
3	M 9	M 20	M 5*	M5*	M 5
4	M 1*	M 1*	M 20	M 20	M 20
5	M 5*	M 5*	M 1*	M 5	M 5*
6	M 5	M 5	M 5	M 1	M 1*
7	M 1	M 1	M 1	M 1*	M 1

An autoregressive model with exogenous inputs (ARX) (See Section 3.4.2) was developed to evaluate the performance of the best ANFIS models (Model 9, 9*, and 20) obtained from this study. The ARX parameters n_a and n_k (See Section 3.4.2) were varied from 0 to 5 and n_b was varied from 1 to 6. Similar to ANFIS models, Event 3 was employed for calibration and Events 5, 7, and 8 for verification. Model parameters were estimated for each of 216 possible combinations between n_a , n_b , and n_k using the MATLAB Identification Toolbox. As mentioned earlier in Section 3.4.2, following Hsu et al. (1995), model fit was evaluated using three residual statistics: *RMSE*, Akaike Information Criterion (*AIC*) (Akaike, 1974), and Bayesian Information Criterion (*BIC*) (Rissanen, 1978). Based on this evaluation criteria, the ARX(2,2,1) model was found to be the best amongst models that included both rainfall and previous discharge as inputs, and the ARX(0,6,3) model was found to be the best among ARX models that used rainfall antecedents only ($n_a = 0$) as input. Moreover, a Naïve model (see Section 3.4.1) was also developed to predict $Q(t)$, $Q(t+1)$, $Q(t+2)$, $Q(t+4)$, and $Q(t+8)$. The values of CE , r^2 , $RMSE$, MAE , RPE and δ obtained from forecast predictions by the naïve model and selected ARX models are compared with best ANFIS model (Model 9, 9*, and 20) results in Table 4.8. It can be seen that ANFIS model 9* shows comparable results for discharge forecasts $t_l < t+2$ when compared to Naïve model and slightly worse results when compared to ARX(2,2,1). For $Q(t+2)$ and $Q(t+4)$, Model 9* is superior to ARX(2,2,1) and Naïve

model. However, Model 9* fails to perform well for $Q(t+8)$ compared to ARX(2,2,1) and the Naïve model. Model 9 which used rainfall inputs only, outperforms ARX(0,6,3) and Naïve model for $t_l > t$. In general, Models 9 and 9* were superior to other models in terms of time shift error δ for $t_l > t$. Due to the high correlation between $Q(t-1)$ (as input) and $Q(t)$ and $Q(t+1)$ (as output) it is expected to see high performance in Model 20, Naïve model, and ARX(2,2,0). However, for longer lead times, the performances of the afore-mentioned models deteriorate due to the recursive nature of the model. Moreover, ARX models show slightly better performance in peak estimation compared to the two ANFIS models. Naïve model in which $Q(t-1)$ time series is considered as the simulated discharge time series for discharge forecasts ($Q(t)$, $Q(t+1)$, $Q(t+2)$, $Q(t+4)$, and $Q(t+8)$) shows $RPE = 0$ for all forecasts. However, the large time shift error for longer lead times is the drawback of its peak estimation capability.

Figures 4.13(a), 4.13 (b), 4.13 (c), 4.13 (d), and 4.13 (e) illustrate the plots of measured discharge and the simulated discharge by Models 9 and 9* for one of the typical testing events, Event 2, in prediction of $Q(t)$, $Q(t+1)$, $Q(t+2)$, $Q(t+4)$, and $Q(t+8)$, respectively. The hydrographs simulated by Models 9 and 9* show good agreements with the measured hydrographs both in general shape and estimated peak discharges. Model 9 however, predict hydrographs with some inaccuracies with a small but noticeable first peak at around 2 min which is in response to the initial rainfall which is actually abstracted. This can be considered as a drawback of the model as it is unable to account for the initial loss. However, Models 9* which benefits from having an antecedent discharge in its inputs show better performance in this regard compared to Model 9 which used only antecedent rainfall inputs. It should be mentioned that in longer lead time Model 9* also shows noticeable first peak at around 2 min which could be attributed to the lower correlation between $Q(t-1)$ (input) with $Q(t+8)$ (output).

Table 4.8: Comparison between Naïve, ARX, and ANFIS models in Catchment 1.

Model	CE	r^2	$RMSE$ (Ls^{-1})	MAE (Ls^{-1})	RPE	δ (sec)
Q(t)						
ANFIS 9	0.950	0.957	0.034	0.022	0.060	3
ANFIS 9*	0.988	0.989	0.019	0.010	0.055	7
ANFIS 20	0.985	0.985	0.022	0.012	0.006	15
Naïve	0.984	0.985	0.022	0.012	0	15
ARX(2,2,1)	0.996	0.996	0.011	0.007	0.020	0
ARX(0,6,3)	0.944	0.952	0.040	0.023	0.090	10
Q(t+1)						
ANFIS 9	0.950	0.957	0.034	0.022	0.060	3
ANFIS 9*	0.966	0.971	0.028	0.015	0.046	10
ANFIS 20	0.946	0.947	0.040	0.022	0.022	30
Naïve	0.946	0.946	0.041	0.022	0	30
ARX(2,2,1)	0.975	0.975	0.027	0.015	0.020	15
ARX(0,6,3)	0.924	0.932	0.049	0.027	0.090	23
Q(t+2)						
ANFIS 9	0.950	0.957	0.034	0.022	0.060	3
ANFIS 9*	0.955	0.964	0.032	0.018	0.060	10
ANFIS 20	0.895	0.896	0.056	0.031	0.041	45
Naïve	0.892	0.895	0.057	0.030	0	45
ARX(2,2,1)	0.927	0.929	0.048	0.025	0.039	37
ARX(0,6,3)	0.885	0.896	0.060	0.033	0.090	38
Q(t+4)						
ANFIS 9	0.934	0.944	0.045	0.025	0.120	10
ANFIS 9*	0.881	0.894	0.052	0.028	0.092	20
ANFIS 20	0.769	0.776	0.083	0.049	0.090	75
Naïve	0.755	0.774	0.086	0.046	0	75
ARX(2,2,1)	0.814	0.824	0.076	0.042	0.053	68
ARX(0,6,3)	0.770	0.793	0.085	0.046	0.090	68
Q(t+8)						
ANFIS 9	0.752	0.786	0.091	0.051	0.198	47
ANFIS 9*	0.305	0.547	0.150	0.081	0.126	24
ANFIS 20	0.507	0.550	0.120	0.079	0.217	135
Naïve	0.457	0.550	0.127	0.073	0	135
ARX(2,2,1)	0.537	0.599	0.119	0.069	0.106	128
ARX(0,6,3)	0.483	0.574	0.124	0.071	0.090	128

Figure 4.13: Plots of observed discharge versus the best and the worst simulated hydrographs of Event 2 for (a) $Q(t)$, (b) $Q(t+1)$, (c) $Q(t+2)$, (d) $Q(t+4)$, and (e) $Q(t+8)$.

4.9 Conclusions

The following can be concluded from this study:

- (i) Among different ANFIS models developed in this study, the models that used the pruned sequential rainfall, non-sequential rainfall and the model which used the most recent discharge antecedent as the only input performed better compared to other models. In general however, models using only rainfall antecedents as inputs were found to be superior in long-term discharge forecasts while models which included $Q(t-1)$ as input were better in short-term discharge predictions.
- (ii) Discharge antecedents are a source of time shift error in discharge forecasts; therefore, models using rainfall only showed smaller time shift errors compared to those that contain $Q(t-1)$ in the input. In addition, using a non-sequential rainfall time series was found to be a suitable way in reducing time shift errors for all ranges of discharge predictions.
- (iii) The overall performance of the ANFIS models quantified via a ranking procedure revealed that using non-sequential rainfall time series (Model 9 and 9*) gave the best performance in terms of goodness-of-fit, peak estimation, and time shift error. The model which used the sequential time series as input was found to give the worst results. Using discharge as the only input gave good results only for short-term predictions; the overall performance deteriorates for longer lead times.
- (iv) ANFIS models 9 and 9* showed comparable if not superior results when compared with ARX models (2, 2, 1) and (0, 6, 3), respectively. However, ARX models showed slightly better performance in peak estimation compared to the two ANFIS models.
- (v) ANFIS models 9 and 9* showed better results when compared with Naïve model for all lead times except for $Q(t)$ in Model 9 and $Q(t+8)$ in Model 9* where Naïve model performed better than ANFIS models.

CHAPTER 5 TRAINING EVENT SELECTION FOR EVENT-BASED RAINFALL-RUNOFF MODELING BY ANFIS

5.1. Introduction

As was discussed in Chapter 2.3, the influence of physical characteristics of hydrograph in training event selection for event-based rainfall-runoff modeling has not been addressed sufficiently in literature yet. Therefore, the objective of the present study was to investigate the influence of hydrograph shape and lag time on the performance of data-driven models in event-based R-R modeling. The ANFIS model was chosen for this study because it encompasses both the learning capabilities of neural networks and the reasoning capabilities of fuzzy inference systems. The 66 events (Appendix B) of Catchment 2 (Fig. 3.15) are used for this analysis. This study is believed to be the first to demonstrate the hydrograph shape and lag effect for event-based modeling using data driven models in a real catchment.

5.2. Influence of Hydrograph Shape on Event-Based Rainfall-Runoff Modelling

5.2.1 Selection of Training Event Candidates

Data driven models (such as ANFIS) as well as physically-based models (such as SWMM) need to be calibrated by a portion of data set known as calibration data. The criterion for choosing training/calibrating events is different between SWMM and ANFIS models, due to the difference in modeling paradigms. SWMM model (Section 3.3.2) employs the rainfall-runoff time series of its calibration data set to tune the 9 physical parameters of the model (see Section 3.3.2). Tan et al., (2008) calibrated a SWMM model for Catchment 2 by using 10 representative events (Events 16 Apr 05, 06 May 05, 16 May 05, 21 May 05, 25 May 05, 26 Jul 05, 05 Oct 05, 16 Oct 05, 24 Dec 05, and 15 Apr 06) selected based on a correlation between the direct runoff volume and total rainfall volume. This SWMM model is known as model SWMM1 in this study. As a first attempt to compare the ANFIS model results to SWMM1, the same 10 events were considered as the candidates to

be used as training events in the ANFIS model. This ANFIS model will be denoted as ANFIS1 in this study.

The selection of events for ANFIS should be based on the mapping of rainfall time series to the discharge time series. To facilitate this process, the hydrograph shape of all 66 events were reviewed and it was found that the hydrographs could be categorized under three general groups: (1) Group 1 - One peak in rainfall and one peak in runoff, (2) Group 2 - Multiple peaks in rainfall and one peak in runoff, and (3) Group 3 - Multiple peaks in rainfall and multiple peaks in runoff. Events 19 Jul 05, 24 Oct 05, and 06 Apr 06 which are typical of events under Groups 1, 2, and 3, respectively are shown in Fig. 5.1.

Figure 5.1: Typical time series of rainfall and discharge for (a) Group 1, (b) Group 2, (c) Group 3 events in Catchment 2.

The Appendix B lists the grouping for all 66 events of Catchment 2 used in this study. Next, three representative events with a wide range of peak direct runoff were selected from each group. The 9 events were categorized into 3 training sets named S1, S2, and S3 which were selected from Group 1, Group 2, and Group 3, respectively. These events are summarized in Table 5.1. Care was taken in the selection of S1, S2, and S3 such that only regular events (with moderate peak flow) were considered in order to avoid the model being trained by extreme events. The ANFIS model developed based on these training event candidates was denoted as ANFIS2.

Table 5.1: Selected training/verification events in ANFIS2 model in Catchment 2.

Training Set	Included Events	Group
S1	19 Jul 05, 30 Jul 05, 20 Aug 05	1
S2	16 May 05, 05 Oct 05, 17 Jan 06	2
S3	26 Jul 05, 16 Oct 05, 03 Nov 06	3

5.2.2. Selection of Input Candidates

Following the same approach discussed in Section 4.2 and using the aforementioned set of calibration events chosen for ANFIS1 and ANFIS2, a list of input candidates (sequential, pruned sequential, and non-sequential) was resulted by Correlation analysis (CA), Mutual Information-Correlation Analyses (MICA), and stepwise regression (SR). The cross-correlation analysis between rainfall and discharge time series revealed that the higher correlation coefficient for calibration data of ANFIS1 and ANFIS2 are obtained for the lag times (t-5), (t-6), and (t-7). Therefore, the time window for sequential rainfall inputs is considered up to (t-6). Table 5.2 summarizes the input candidates for ANFIS1 and ANFIS2 models. In the following section, the final training events and inputs will be determined through a verification process.

Table 5.2: Input candidates selected based on CA, MICA, and SR methods for ANFIS1 and ANFIS2 models in Catchment2.

Candidate	No. of Inputs	Inputs (ANFIS1)	Inputs (ANFIS2)	Remark
C1	7	$R(t), R(t-1), \dots, R(t-6)$	$R(t), R(t-1), \dots, R(t-6)$	CA
C2	6	$R(t), R(t-1), \dots, R(t-5)$	$R(t), R(t-1), \dots, R(t-5)$	CA
C3	3	$R(t-4), R(t-5), R(t-6)$	$R(t-4), R(t-5), R(t-6)$	CA
C4	3	$R(t-5), R(t-6), R(t-7)$	$R(t-5), R(t-6), R(t-7)$	CA
C5	3	$R(t-6), R(t-7), R(t-8)$	$R(t-6), R(t-7), R(t-8)$	CA
C6	2	$R(t-4), R(t-8)$	$R(t-3), R(t-8)$	MICA
C7	2	$R(t-3), R(t-7)$	$R(t-4), R(t-8)$	MICA
C8	2	$R(t-3), R(t-8)$	$R(t-3), R(t-9)$	MICA
C9	3	$R(t-3), R(t-5), R(t-8)$	$R(t-3), R(t-6), R(t-9)$	MICA
C10	3	$R(t-3), R(t-6), R(t-9)$	$R(t-4), R(t-6), R(t-9)$	MICA
C11	3	$R(t-2), R(t-5), R(t-8)$	$R(t-4), R(t-5), R(t-8)$	MICA
C12	4	$R(t-1), R(t-3), R(t-6), R(t-8)$	$R(t), R(t-3), R(t-6), R(t-9)$	MICA
C13	4	$R(t-2), R(t-4), R(t-7), R(t-9)$	$R(t-2), R(t-3), R(t-7), R(t-9)$	MICA
C14	4	$R(t), R(t-3), R(t-5), R(t-8)$	$R(t-2), R(t-5), R(t-8), R(t-11)$	MICA
C15	7	$R(t-2) \dots R(t-7), R(t-9)$	$R(t-2) \dots R(t-7), R(t-10)$	SR

5.2.3 ANFIS Models Development

Among the 10 training event candidates for ANFIS1, five (Events 06 May 05, 16 May 05, 26 Jul 05, 16 Oct 05, and 24 Dec 05) were chosen as training events while the remaining 5 were used for verification. All 5 selected training events had moderate peak discharge and this was to avoid the being trained with extreme events only. The 15 input candidates were used in a verification analysis by ANFIS1 and the results revealed that using C7 input combination gave the highest performance in simulating verification events followed by C6 and C9. Therefore, $R(t-3)$ and $R(t-7)$ were applied to ANFIS1 to test all 66 events in the testing phase.

In order to obtain the appropriate combination of training event candidates from the three sets (Table 5.1) for the training and verification of the ANFIS2, a three-fold cross validation analysis was adopted; i.e. two sets were chosen as training and the third one for verification. This was repeated for all possible combinations of training and verification sets. For each combination, the average CE, r^2 , RMSE, MAE, and RPE values over the 15 input candidates are calculated and compared to find the best training set. Table 5.3 summarizes the results of cross-validation procedure. As can be seen, applying S1 and S2 as training datasets and S3 as the verification dataset was found to give the best combination. It was concluded that those events which had one peak in the runoff hydrograph (with either one peak or multi peaks in rainfall) were the most suitable for ANFIS model training.

Table 5.3: Verification results obtained by ANFIS2 trained with different training sets in Catchment 2.

Training Set No.	Verification Set No.	<i>CE</i>	r^2	<i>RMSE</i>	<i>MAE</i>	<i>RPE</i>
S ₁ , S ₂	S ₃	0.810	0.826	1.839	1.287	0.104
S ₁ , S ₃	S ₂	0.442	0.789	2.141	1.402	0.293
S ₂ , S ₃	S ₁	0.695	0.822	2.564	1.895	0.189

After finding the proper training set, the performance of each of 15 input candidates in terms of CE, r^2 , RMSE, MAE, and RPE values were compared. Results showed that using C6 gives the highest performance statistics followed by C8 and C7. Therefore, a combination of R(t-3) and R(t-8) was applied in ANFIS2 to test all 66 events in the testing phase. In general, the verifications results of ANFIS 1 and ANFIS2 during input selection phase revealed that using non-sequential rainfall inputs gave better performance compared to pruned sequential and sequential rainfall inputs. This is consistent with the similar findings on Catchment 1 which were discussed in Chapter 4.

In developing ANFIS1 and ANFIS2, the initial epoch value was taken as 30. However, after finalizing the training events and inputs of the models a sensitivity analysis (similar to Section 4.5) was done to achieve the best choice of epoch to avoid over fitting. Results showed that using 30 to 50 epochs would be a good option since for the bigger epoch values verification performance started to decrease while the training performance improved. This was considered as the evidence of over fitting. Therefore, epoch = 40 was adopted for testing phase of both models.

Moreover, initially two Gaussian membership functions were assigned for each input. After finalizing the training events and inputs of the models a sensitivity analysis (similar to Section 4.4) was conducted to find the best choice of number and type of membership functions. The results revealed that both models perform slightly better when a triangular membership function is used. Increasing the number of membership function not only didn't improve the performance but also gradually deteriorated the models' performances. Therefore, two triangular membership functions were assigned for testing phase.

5.2.4 SWMM Models Calibration

As mentioned in Section 5.2.1, SWMM1 was calibrated by the 10 events selected based on a correlation between the direct runoff volume and total rainfall volume (Tan et al., 2008). After calibrating the SWMM1 with these 10 events, a set of 9 calibration parameters was obtained which was found to give reasonable estimates to the runoff hydrograph for all 66 events. The average values of these parameters are summarized in Table 5.4.

Table 5.4: SWMM1 calibration: Estimated parameters and performance.

Parameters									Performance	
W	Imp	S_0	$Nimp$	$Nper$	Dp	f_o	f_c	K	CE	RPE
(m)	(%)	(m/m)	($m^{-1/3}_s$)	($m^{-1/3}_s$)	(mm)	(mm/h)	(mm/h)	(l/s)	-	-
2 620	27.8	0.0073	0.0106	0.6	7	143.3	25	0.000389	0.8974	0.064

A new SWMM model known as SWMM2 was next calibrated using the same 9 events used for calibrating ANFIS2 (Table 5.1). SWMM2 was calibrated using exactly the same 9 events as ANFIS in order to test the choice of calibration events on model performance between the two methods. The procedure for calibration was similar to model SWMM1. Note that only four of the events in SWMM2 (Events 16 May 05, 26 Jul 05, 05 Oct 05, 16 Oct 05) were previously used in SWMM1. The average values of calibration parameters for SWMM2 are summarized in Table 5.5. Comparing the calibration parameters presented for SWMM1 and SWMM2 in Tables 5.4 and 5.5, respectively, model calibration did not change significantly between SWMM1 and SWMM2. It can therefore be inferred that SWMM model calibration is insensitive between these two sets of calibration events.

Table 5.5: SWMM2 calibration: Estimated parameters and performance.

Parameters									Performance	
W	Imp	S_0	$Nimp$	$Nper$	Dp	f_o	f_c	K	CE	RPE
(m)	(%)	(m/m)	($m^{-1/3}_s$)	($m^{-1/3}_s$)	(mm)	(mm/h)	(mm/h)	($1/s$)	-	-
2 699.4	22.3	0.0072	0.0135	0.6	7	143.3	25	0.000389	0.8996	0.136

5.2.5 Results and Discussion

The SWMM1, ANFIS1, SWMM2, and ANFIS2 models were employed to simulate all the 66 storm events of Catchment 2 listed in the Appendix B and individual values of CE , r^2 , $RMSE$, MAE , and RPE were calculated. The average values of CE , r^2 , $RMSE$, MAE , and RPE are presented in Table 5.6 as a measure of the gross performance of the SWMM1, ANFIS1, SWMM2, and ANFIS2 models. It can be observed from Table 5.6 that the SWMM models are not sensitive to the change in calibration events since the overall performance of SWMM1 and SWMM2 are similar (Note that the number of calibrating events are approximately the same for SWMM1 and SWMM2 models). The performance of the ANFIS models, however, is significantly affected by changes to the events used in training. It is evident that the response of the catchment to different rainfall events in terms of hydrograph shapes would be an appropriate paradigm to use in event selection for calibration,

since ANFIS learns by association. This makes ANFIS models more “sensitive” to event selection in model calibration. Considerable improvements to CE (approximately 80% increase) and RPE (approximately 25% decrease) and moderate improvement in r^2 (2% increase), $RMSE$ (6% decrease), and MAE (2% decrease) for ANFIS2 compared to ANFIS1 were obtained.

Table 5.6: Average values of evaluation metrics over 66 events of Catchment 2 simulated by different models.

Models	CE	r^2	$RMSE$	MAE	RPE
SWMM1	0.634	0.788	2.641	1.708	0.313
ANFIS1	0.288	0.676	3.339	2.023	0.342
SWMM2	0.628	0.772	2.670	1.743	0.372
ANFIS2	0.514	0.690	3.126	1.976	0.262

A comparison of the distribution of CE values between the two groups of models is made in Fig. 5.2. The general distributions in results for all four models look similar; except for the larger number of negative CE values in ANFIS1. The high number of negative CE values for ANFIS1 indicates that selecting training events based on correlation analysis between rainfall and runoff volume may not be a good approach for ANFIS, although it may be appropriate for SWMM. Through a judicious choice of the training events, the number of events with negative CE values is decreased dramatically, from 10 for ANFIS1 to 3 for ANFIS2. This improvement is also evident in the cumulative frequency curves shown in Fig. 5.2. Each value on this curve shows the number of events that have the CE values larger than a specific value. It is observed that the cumulative frequency curves for ANFIS2, SWMM1, and SWMM2 are located close together while ANFIS1 curve is located below. Thus, a change in the selection of training events for the ANFIS model improves its performance making it comparable to that of SWMM. As seen from the cumulative curves, the models are able to achieve CE of 0.7 or higher approximately 50% of the time.

Figure 5.2: Events distribution in different intervals of CE for proposed models.

Model results of the best performing ANFIS (ANFIS2) and SWMM (SWMM1) models are compared in order to directly evaluate the capabilities of ANFIS relative to SWMM. The CE values of all events simulated by SWMM1 and ANFIS2 (with the exception of Events 12 Mar 05, 27 Mar 05, and 21 Jan 06 which had negative CE values) are compared in a scatter plot in Fig. 5.3. Each group of events (See section 5.2.1) has been presented with a different symbol in order to distinguish between the models in simulating the events within the different groups. Figure 5.3 shows that there is an approximately equitable distribution of data points about the perfect agreement line. 28 events or approximately 42% of events had higher CE values when modeled by ANFIS2 compared to SWMM1. 21 out of these 28 events were either from Group 1 or Group 2. It was also found that the ANFIS2 model ability in simulating Group 1 events was better than SWMM1, as 6 out of 9 of these events were located above the perfect agreement line. Considering a margin of error of $\pm 30\%$, 4 events from Group 1 are modeled better by ANFIS2 than SWMM (See Fig. 5.3).

Figure 5.3: Scatter plot of CE values for events simulated by SWMM1 and ANFIS2.

Figure 5.4 illustrates the observed and simulated hydrographs for Event 30 Dec 04, one of the events from Group 1 that were modeled better by ANFIS2 than SWMM1. As can be seen, ANFIS2 is more successful in simulating the general shape of the hydrograph and peak flow compared to SWMM1 which shows an early rising limb, late recessing limb, and underestimation of the peak discharge. Further, it is observed that 7 Group 3 events were modeled better by SWMM1 than ANFIS2, while only 3 Group 3 events were modeled better by ANFIS2 in Fig. 5.3. The relatively worse performance by ANFIS in modeling Group 3 events is probably because Group 3 events were not included in the ANFIS training dataset. For example, Event 05 Jan 06, one of the events from Group 3 which were modeled better by SWMM1 than ANFIS2 is selected and the observed and simulated hydrographs are shown in Fig. 5.5. It can be inferred that rising and recessing limbs are modeled better by SWMM1; however, both models suffer from underestimation of runoff peak, although the underestimation is more acute for ANFIS2.

Figure 5.4: Observed and simulated hydrographs of Event 30 Dec 04 (*CE* for SWMM1 and ANFIS2 are 0.466 and 0.813, respectively).

Figure 5.5: Observed and simulated hydrographs of Event 05 Jan 06 (*CE* values for SWMM1 and ANFIS2 are 0.843 and 0.462, respectively).

A scatter plot of the *RPE* values is shown in Fig. 5.6 for all 66 events in order to evaluate the ability of both models in peak flow estimation. As can be observed, the larger number of data points below the perfect agreement line suggests poorer peak flow estimation by SWMM in comparison with ANFIS and that the overall ability of ANFIS in peak flow estimation is generally superior to SWMM. Among the 66 events simulated in this study, *RPE* values for 23 events

simulated with ANFIS2 were less than 0.15 while in SWMM1 only 14 events had values less than 0.15. In addition, 37 events had smaller values of RPE when modeled with ANFIS compared to SWMM. It was found that the ANFIS2 model was superior in peak flow estimation especially for Group 1 and Group 2 events. 6 out of 9 of events in Group 1, 19 out of 34 events in Group 2, and 12 out of 22 events in Group 3 were better estimated by ANFIS2 than SWMM1.

Figure 5.6: Scatter plot of RPE values for events simulated by SWMM1 and ANFIS2.

A scatter plot of CE versus RPE for the SWMM1 and ANFIS2 models is shown in Fig. 5.7. A good R-R model is expected to perform well in simulating both the runoff hydrograph shape and peak flow (high CE and small RPE values). If 0.75 and 0.15 are considered as the arbitrary minimum and maximum required values for CE and RPE , respectively for a good simulation, it can be inferred that both SWMM1 and ANFIS2 models are not able to provide good estimates of both hydrograph shape and peak flow simultaneously for all events, since only a few events fall within the requirement of a good R-R model. However, it can be seen that in all three groups of events the overall performance of ANFIS2 model is comparable to SWMM1. ANFIS is then able to emulate the catchment response to

rainfall as well as a conventional physically-based model, although both models are not able to make perfect simulations.

Figure 5.7: Scatter plot of CE versus RPE for events simulated by SWMM1 and ANFIS2: (a) Group 1, (b) Group 2, and (c) Group 3.

5.3 Influence of Lag Time on Event-Based Rainfall-Runoff Modelling

5.3.1 Lag time and selection of training events

The lag time is an important concept used in hydrologic modeling and design. For example, the Snyder unit hydrograph synthesis method (Snyder, 1938) requires an estimate of lag time. Several definitions are available for lag time in the literature. Two of the most widely used definitions are: (1.) Time interval between the centroid of the effective rainfall and the centroid of direct runoff (t_1); and (2.) Time interval between the centroid of effective rainfall and peak of direct runoff (t_2) (Fig. 5.8). In any rainfall-runoff event, this lag time could be different due to differences in initial soil moisture, rainfall intensity and other factors (Singh, 1988).

Figure 5.8: Schematic representation of lag time based on two different definitions.

We adopted cross-correlation analyses between the rainfall and runoff time series to determine the lag time for an individual event. The lag time is thus defined as the number of time shifts (τ_m) required to provide the highest cross-correlation coefficient. This method of determining the lag time is not entirely consistent with

the traditional definition of lag time mentioned earlier, but can be considered to provide an approximation of the actual lag time. From the 66 events available (Appendix B) only 62 were used in this study since the remaining 4 events (12 Mar 05, 27 Mar 05, 21 Jan 06, and 31 Oct 06) were considered to be atypical. The lag time for all 62 events considered in this study were determined from cross-correlation analysis and the distribution of τ_m for the events are shown in Fig 5.9. It is observed that τ_m varies between 15 to 55 minutes and the average and median τ_m are 33 and 30 minutes, respectively. A majority (47 or approximately 76%) of the events have τ_m values within the range of 25 to 40 minutes.

Figure 5.9: Distribution of 62 events over different lag time.

An initial analysis was performed on the events where the lag times were predominant. Thus, 47 events with lag times of 25 min, 30 min, 35 min and 40 min (Fig. 5.9), respectively were segregated in four groups: G_5 , G_6 , G_7 , and G_8 . The number of events in each of these groups is 15, 11, 9 and 12, respectively. The events used for training the ANFIS model were selected based on 2 criteria: lag time and hydrograph shape. Thus, the dataset was divided into four training datasets: S_5 , S_6 , S_7 , and S_8 . Each of the training sets comprised three events, selected from groups G_5 , G_6 , G_7 , and G_8 , respectively. A sensitivity analysis was conducted

to find the minimum number of training events required. It was revealed that at least 3 events should be considered in each training group. Table 5.7 shows the training events and details of the selected events can be found in Appendix B. Following the findings in Section 5.2, care was taken to ensure that the three events selected for each of the training datasets comprised single peak events, i.e. events which exhibited single peaks in the hydrographs. In addition, events with moderate peak discharge were preferred to avoid training the model with extreme events. The ANFIS models developed based on the mentioned training sets are denoted as S_iG_j where S_i ($i = 5, 6, 7$, and 8) refers to a training set and G_j ($j = 5, 6, 7, 8$) refers to a testing set. Thus, as an example, S_5G_6 means an ANFIS model trained with events from S_5 and tested with events from G_6 . Given that there are 4 groups of events each in S_i and G_j , a total of sixteen models were thus developed.

Table 5.7: Training events used for each group.

Training Set	Group	Events
S5	G5	23-Aug-05, 31-Dec-05, 8-Apr-06
S6	G6	19-Jul-05, 30-Jul-05, 24-Dec-05
S7	G7	16-May-05, 2-Aug-05, 18-Sep-05
S8	G8	21-May-05, 26-May-05, 17-Jul-05

5.3.2 Selection of inputs

The input selection procedure is conducted on the same catchment in Section 5.2.2 revealed that the best performing input combinations are non-sequential rainfall inputs which are obtained by MICA (see Section 3.9.3). Therefore, MICA was employed to find the 3-highest ranked combination of 2, 3, and 4 rainfall inputs for each model their input candidates. For each model, the candidate that gave the highest CE value was chosen. Table 5.8 shows the optimal combination of the selected rainfall inputs determined for each ANFIS model.

Table 5.8: Optimal combination of two rainfall inputs for models used in simulating $Q(t)$.

Model	Rainfall Inputs	Model	Rainfall Inputs
S5G5	R(t-3), R(t-7)	S5G7	R(t-3), R(t-7)
S6G5	R(t-4), R(t-9)	S6G7	R(t-3), R(t-8)
S7G5	R(t-5), R(t-12)	S7G7	R(t-2), R(t-7)
S8G5	R(t-3), R(t-8)	S8G7	R(t-2), R(t-8)
S5G6	R(t-6), R(t-12)	S5G8	R(t-4), R(t-8)
S6G6	R(t-5), R(t-10)	S6G8	R(t-4), R(t-8)
S7G6	R(t-6), R(t-12)	S7G8	R(t-5), R(t-8)
S8G6	R(t-2), R(t-6)	S8G8	R(t-2), R(t-9)

5.3.3 Results and discussions

5.3.3.1 Effect of lag time

Figures 5.10, 5.11 and 5.12 show box and whisker plots for CE , r^2 and $RMSE$ values obtained from simulations of the test events by the 16 ANFIS models. The highest median CE values in Figs. 5.10(a), (b), (c), and (d) were obtained by the models S_5G_5 , S_6G_6 , S_8G_7 , and S_8G_8 , respectively. In general, the best ANFIS models are those where the training and testing events were chosen from groups with similar lag time. Furthermore, it is observed from the figures that model performance in general, worsens as the difference in lag time between the events in the training and testing events diverge; the greater the divergence, the poorer the fit of the model to the measured data.

On the whole, the plots for r^2 and $RMSE$ are consistent with these observations. Hence, in Fig. 5.10(a) the highest median CE value ($CE = 0.766$) was obtained by S_5G_5 compared to median CE values of 0.641, 0.421, and 0.476 obtained by models S_6G_5 , S_7G_5 , and S_8G_5 , respectively. Although the general trends

mentioned earlier are clear, a few exceptions are apparent. Firstly, S_7G_8 in Fig. 5.10(d) does not show a continuously decreasing trend as expected. Secondly, S_7G_7 in Fig. 5.10(c) should have the maximum value but is marginally less than S_8G_7 . The last exceptions are S_5G_7 in Fig. 5.11(c), which is expected to give lower r^2 value compared to S_6G_7 , and model S_6G_8 in Fig. 5.12(d) which shows a lower median $RMSE$ compared to S_7G_8 . These discrepancies in the results can be attributed to other subtleties associated with the hydrograph characteristics (such as the gradient, differences between the rising and falling limbs) which could have an impact, but were not considered in this study.

Figure 5.10: CE for different models used to simulate $Q(t)$ for events of group (a) G5, (b) G6, (c) G7, and (d) G8.

Figure 5.11: r^2 for different models used to simulate $Q(t)$ for events of group (a) G5, (b) G6, (c) G7, and (d) G8.

Nevertheless, the results from this analysis show that the model is sensitive to the lag time of events used in training, and generally speaking, model performance deteriorates as the lag time between the events used in the training and testing dataset increases. This is to be expected since data driven models learn by association. Thus, when the lag times of events used for training are different from the lag times of events used in the testing phase, there would be a mismatch in the phasing of the input-output relationship and this effect becomes more significant as the difference in lag time increases.

Figure 5.12: *RMSE* for different models used to simulate $Q(t)$ for events of group (a) G5, (b) G6, (c) G7, and (d) G8.

5.3.3.2 Model development based on lag time

Previously in Section 5.2, two ANFIS models were developed using the same dataset; one model trained with events that were chosen without consideration of the hydrograph shape or lag time (denoted as ANFIS1) and the second model trained with events that were selected based on hydrograph shape (denoted as ANFIS2). The events used in the training of the ANFIS1 and ANFIS2 models are shown in Figs. 5.13 and 5.14, respectively. Note that lag time was not a consideration in the choice of the training events shown in Fig. 5.13 and 5.14.

Figure 5.13: Time series of rainfall and discharge used as training events for ANFIS1: (a) 06 May 05, (b) 16 May 05, (c) 26 Jul 05, (d) 16 Oct 05, and (e) 24 Dec 05.

Figure 5.14: Time series of rainfall and discharge used as training events for ANFIS2: (a) 19 Jul 05, (b) 30 Jul 05, (c) 20 Aug 05, (d) 16 May 05, (e) 05 Oct 05, (f) 17 Jan 06.

A comparison is made here where the ANFIS models developed in this study (training events are selected based on both hydrograph shape and lag time) are compared against the ANFIS1 and ANFIS2 models. In order to facilitate this comparison, sub-models differentiated by lag times were developed. Rather than creating one sub-model each for events with a specific lag time, which would be impractical, an integrated approach was adopted; events from contiguous groups of lag times were incorporated into three separate groupings. Three sub-models: ANFIS3a, ANFIS3b, and ANFIS3c corresponding to events having lag times within the [15, 20], [25, 45], and [45, 55] (min) time intervals, respectively, were thus developed. The inputs to these models were determined based on the methodology outlined in Section 5.3.2. ANFIS3a, ANFIS3b and ANFIS3c models were used to simulate events with the lag times: $\tau_m \leq 20$ min. (6 events), $25 \text{ min.} \leq \tau_m \leq 40$ min. (47 events) and $\tau_m \geq 45$ min. (9 events), respectively. Details of the training and verification events chosen for the ANFIS3a, ANFIS3b, and ANFIS3c are provided in Table 5.9 and the hydrographs of the training events are shown in Fig. 5.15. The number of events used in the testing phase of the ANFIS3a, ANFIS3b, and ANFIS3c models including the training and verification events (consistent with Tan et. al.'s (2008) study where the training and verification events were also included) were 6, 47, and 9, respectively. Following the ANFIS2 model, the events in Fig. 5.15 were chosen either from Group 1 or Group 2.

Table 5.9: Calibration events used in each ANFIS3 sub-models.

Sub-Models	Training Events	Lag Time (min)	Verification Events	Lag Time (min)
ANFIS3a	29 Jul 06	15	08 Dec 05	20
	15 Apr 06	20	-	-
ANFIS3b	19 Jul 05	30	17 Sep 06	25
	30 Jul 05	30	04 Jul 05	30
	24 Dec 05	30	16 May 05	35
	-	-	06 Apr 06	40
ANFIS3c	05 Oct 05	50	14 Apr 06	45
	05 Sep 06	50	01 Jun 05	50

Figure 5.15: Time series of rainfall and discharge used as training events for ANFIS3: (a) 29 Jul 06, (b) 15 Apr 06, (c) 19 Jul 05, (d) 30 Jul 05, (e) 24 Dec 05, (f) 05 Oct 05, and (g) 05 Sep 06.

Detailed results of the CE , r^2 , $RMSE$, and RPE values for the ANFIS1, ANFIS2 and ANFIS3 models are compared in Figs. 5.16(a) – (d), respectively. In general, the ANFIS3 sub-models perform significantly better than the ANFIS1 and ANFIS2 models. The median CE and r^2 values of the ANFIS3a, ANFIS3b and ANFIS3c models are consistently higher and represent significant improvements over the performance of the ANFIS1 and ANFIS2 models. However, the median $RMSE$ value for ANFIS3a is higher compared to the other ANFIS models. This is attributed to the fact that $RMSE$ is biased against errors in the simulation of high flow events (peak discharge $> 30 \text{ m}^3/\text{s}$). Out of the 6 events modeled by ANFIS3a, four were high flow events; the $RMSE$ parameter therefore penalized a majority of the events simulated by ANFIS3a, resulting in the high $RMSE$ values. In addition, the median RPE values for the ANFIS3 models are smaller than that for ANFIS1 and ANFIS2, indicating that considerations of the lag effect results in better estimates of the peak discharge. The values of CE , r^2 , $RMSE$, and RPE averaged over all 62 events are reported in Table 5.10. Note that results for ANFIS3 have been averaged over the three sub-models (average of ANFIS3a, ANFIS3b and ANFIS3c). As shown in Table 5.10, CE and r^2 values increase according to the order ANFIS1 $\rightarrow$ ANFIS2 $\rightarrow$ ANFIS3 while $RMSE$ and RPE values show a decreasing trend. In fact, ANFIS3 produces consistently better results compared to the ANFIS1 and ANFIS2 models.

In practical situations however, the lag time is not known. Hence, ANFIS3b was used to simulate all the 62 events since the events used by the ANFIS3b model have lag times equal to the median value for all the events. The results of this analysis, indicated by ANFIS4, are compared with other ANFIS models in Fig. 5.16 and Table 5.10. The results for ANFIS4 in Fig. 5.16 show that there is a significant improvement in ANFIS4 over ANFIS1; however, improvements over ANFIS2 are modest. This is attributed to the fact that the events used in the training of ANFIS2 included 2 of the 3 events used in the training of ANFIS3b, thus producing results that are similar. Moreover, ANFIS2 was trained with more events (6 events) in comparison with ANFIS4 (3 events). Nevertheless, the improvement of ANFIS3b over ANFIS1 highlights the importance of the lag time in the selection of training events. Moreover, the results from a SWMM model (SWMM1 in Section 5.2) and

an ARX model (calibrated with same events used for training ANFIS4) are provided in Table 5.10 as a bench mark for further comparison. Following the same training approach described in Sections 3.4.2, ARX(0,6,4) was the best ARX model that used rainfall antecedents only ($n_a = 0$) as inputs. As can be seen in Table 5.10, results obtained by ANFIS4 are comparable with SWMM while it is outperforming ARX model.

Figure 5.16: (a) CE , (b) r^2 , (c) $RMSE$, and (d) RPE for ANFIS1, ANFIS2, ANFIS3a, ANFIS3b, ANFIS3c models.

Table 5.10: Average values of CE , r^2 , $RMSE$, and RPE for 62 events of Catchment 2 obtained by different models.

Models	CE	r^2	$RMSE$ (m^3/s)	MAE (m^3/s)	RPE
ANFIS1	0.394	0.691	3.176	1.957	0.287
ANFIS2	0.586	0.697	3.032	1.939	0.234
ANFIS3	0.659	0.773	2.554	1.711	0.197
ANFIS4	0.570	0.720	2.970	1.933	0.229
SWMM1	0.642	0.770	2.648	1.719	0.348
ARX(0,6,4)	0.423	0.595	3.248	2.158	0.358

5.4 Extension of the Findings on Catchment 1

The study presented in Section 4.2 on Catchment 2 showed that selecting the training events based on the hydrograph shape is a good strategy. It was shown that events with a single peak in the discharge time series are suitable for use as training events for the ANFIS model. In order to support these findings, the same approach was taken for Catchment 1. Among the 10 events of Catchment 1, two (Events 7 and 9) are single peak events, four (Events 1, 2, 3, and 5) are double peak events, and four (Events 4, 6, 8, and 10) are multiple peak events. Therefore, two representative events with different range of peak discharge were selected from each group. The 6 events were categorized into 3 training sets named S1, S2, and S3 which were selected from different groups. These events are summarized in Table 5.11. In order to obtain the appropriate combination from the three sets (Table 5.11) for the training and verification of the ANFIS model, a three-fold cross validation analysis was adopted; i.e. two sets were chosen as training and the third one for verification. This was repeated for all possible combinations of training and verification sets. Applying S1 and S2 as training datasets and S3 as the verification dataset was found to give the best combination. For each training set, the top three input candidates were selected based on MICA method.

Table 5.11: Selected training/verification events for Catchment1.

Training Set	Included Events	Group
S1	7, 9	Singe Peak Events
S2	1, 2	Double Peak Events
S3	4,6	Multiple Peak Events

Table 5.12 summarizes the results of cross-validation procedure. It should be mentioned that the results are based on the average values of the three highest ranked input candidates. As can be seen in Table 5.12, those events which had one or two peaks in the runoff hydrograph were best for ANFIS model training compared to events with multiple peaks. This finding is consistent with the ones for Catchment 2 (Section 5.2).

Table 5.12: Verification results obtained by ANFIS model trained with different training sets in Catchment 1.

Training Set No.	Verification Set No.	<i>CE</i>	r^2	<i>RMSE</i>	<i>MAE</i>	<i>RPE</i>
S ₁ , S ₂	S ₃	0.914	0.942	0.038	0.024	0.146
S ₁ , S ₃	S ₂	0.906	0.931	0.043	0.026	0.192
S ₂ , S ₃	S ₁	0.909	0.937	0.040	0.026	0.167

Due to the small size of Catchment 1, the effect of the lag time could not be studied.

5.5 Conclusions

The following can be concluded from this study:

- SWMM and ANFIS have different sensitivity to any changes in calibration events. Selecting calibration events with regard to the correlation between rainfall and runoff volume, gave acceptable results in SWMM but not for

ANFIS. On the other hand, selecting training events based on the shape of hydrograph was found to be a good strategy in ANFIS. Specifically, it was concluded that those events which had one peak in the runoff hydrograph (with either one peak or multi peaks in rainfall) were the most suitable for ANFIS model training.

- (ii) Using the three different methods for input selection (Section 3.9) in Catchment 2 revealed that using two non-sequential rainfall inputs gave the best testing performance. This is consistent with the similar findings on Catchment 1 (Section 4).
- (ii) ANFIS model is comparable to SWMM in terms of overall goodness-of-fit. It was found that ANFIS performance in terms of *CE* was better than SWMM in Group 1 events. In addition, ANFIS performance was comparable to SWMM in Group 2 events; however, it was worse in Group 3 events compared to SWMM.
- (iii) ANFIS model is better than SWMM in terms of peak flow estimation. It was found that ANFIS performance in peak flow estimation in terms of *RPE* values was superior specifically in Group 1 and Group 2 events.
- (iv) ANFIS model is comparable to SWMM in satisfying both goodness-of-fit and peak flow estimation criteria, simultaneously. It can be inferred that based on these two criteria both models were not perfect but were comparable for all groups of events.
- (v) Analysis of the events segregated in terms of lag time showed that events tested on models trained using events with the same lag time produced the best results. The worst results were obtained when the lag time of the testing events were significantly different from the lag time of the training events. This analysis showed that lag time is a significant factor in the choice of training and testing events used in data-driven models such as ANFIS.
- (vi) In practice, it is not practical to develop specialized models, i.e. models catered to events with specific lag times, since the lag time of an event is not

known *a priori*. Three models developed based on the events with the most common lag times for the catchment showed that each of the three models outperformed the ANFIS model that was developed without consideration of lag time (ANFIS1 model).

- (vii) Lastly, the ANFIS model developed based on events with the median lag time performed better than the ANFIS1 model. This suggests that events with the median lag times should be chosen as training events in event-based rainfall-runoff modelling.
- (viii) The results of the proposed ANFIS model considering lag time effect was compared with SWMM and ARX models as bench mark. ANFIS was comparable with SWMM while outperformed ARX model.
- (ix) The effect of hydrograph shape was investigated also in Catchment 1. Similar findings consistent with what obtained in Catchment 2 supported this finding. However, the effect of lag time couldn't be investigated due to small size of the experimental catchment.

CHAPTER 6 RAINFALL-RUNOFF MODELING USING A TAKAGI-SUGENO NEURO-FUZZY MODEL WITH LOCAL LEARNING

6.1 Introduction

To-date, there have been several studies carried out where NFS have been employed in rainfall-runoff modeling. The ANFIS (Adaptive Network-based Fuzzy Inference System) model is probably the most commonly used off-line Takagi-Sugeno type NFS in rainfall-runoff modeling studies (Nayak et al., 2004; Nayak et al., 2005b; Aqil et al., 2007; Remesan et al., 2009; Mukerji et al., 2009) .

Common neuro-fuzzy systems employ global learning or batch learning, in which all the parameters of the fuzzy system are optimized off-line. In general, global learning suffers from slower training speed, higher sensitivity to noise, and difficulty in on-line simulation (Hong and White, 2009). Some efforts (Kasabov and Song, 2002; Quah and Quek, 2006) have been made to employ local learning in NFS to overcome the drawbacks of global learning specifically for on-line applications. The Dynamic Evolving Neural-Fuzzy Inference Systems (DENFIS) (Kasabov and Song, 2002) model is a Takagi-Sugeno type FIS which employs the on-line or local learning paradigm. In DENFIS, depending on the position of the input vector in the input space, a fuzzy inference system for calculating the output will be formed dynamically based on m fuzzy rules that had been created during the past learning process.

Despite the appeal of on-line NFS models in R-R modeling, no studies have been done and more efforts are needed to reveal the salient features and capabilities of such models in R-R modeling. The overall aim of this study was to investigate the application of local learning in NFS for a rainfall-runoff modeling application. In order to achieve this, the study was carried out in two stages. Firstly, the local learning NFS was tested in a modeling application for both event-based and continuous R-R modeling for three different scales of catchment sizes ranging from an outdoor experimental facility (25 m²) to a small (5.6 km²) urban catchment and a large (2,275 km²) rural catchment. These catchments are denoted as Catchment 1, 2, and 3, respectively. In the second stage, the local learning NFS was tested in a

forecast application for a large (241.3 km²) rural catchment denoted as Catchment 4 in this study. For the first stage, results from the NFS will be compared with the results obtained from ARX and physically-based models, which are adopted as benchmarks. In addition, the local learning NFS will also be compared with ANFIS, a well-known offline NFS. The objective of this comparison was to ascertain the accuracy of the local learning NFS compared to benchmarks selected. The real time adaptability of the online will be verified in a flow forecast application in Stage 2. Here, results from the online model (with real time adaptability) will be compared against ANFIS for Catchment 4. The objective of this comparison was to ascertain the improvements obtained by the local learning NFS in a forecast application. Moreover, the results are compared with ARX and Naïve models which are adopted as bench marks.

6.2 Dynamic Evolving Neural Fuzzy Inference System (DENFIS)

6.2.1 Evolving clustering method (ECM)

DENFIS modelling evolves through incremental learning resulting in local element adjustment as new data is introduced to the model in sequence. DENFIS utilizes an evolving, online clustering method called the Evolving Clustering Method (ECM) (Kasabov and Song, 2002). ECM is an online, maximum distance-based clustering method implementing a scatter partitioning of the input space to create fuzzy inference rules. It is a fast, one-pass algorithm for a dynamic estimation of the number of clusters in a data set and finding their current centres in the input space. In each cluster, the maximum distance between a presented data point and the cluster centre is less than a threshold value, D_{thr} . This threshold is a clustering parameter and controls the number of clusters to be estimated. In this method, the definition of distance between two vectors x and y is based on *general Euclidian distance* defined as follows:

$$\|x - y\| = \left(\sum_{i=1}^q |x_i - y_i|^2 \right)^{1/2} / q^{1/2} \quad (6.1)$$

where q is the number of input data points and $x, y \in R^q$.

When presented with additional data points, some new clusters may be created or some currently available clusters may be updated through the changing of the centres' positions and increase of the cluster size. The position of the current data point governs which cluster will be updated and how much it should be changed. The updating process of a cluster will be stopped when its cluster radius reaches a threshold value, D_{thr} . Figure 6.1 illustrates the clustering process in 2-D space and the flow chart of the ECM algorithm is shown in Figure 6.2. The ECM algorithm can be summarized as follows:

Step 0: The first cluster C_1^0 is created by assigning the first data point in the input stream as the first cluster centre Cc_1^0 and setting the cluster radius $R_1 = 0$ (Fig. 6.1(a)).

Step 1: If all the data points in the time series have been presented to the algorithm, the algorithm is terminated. Else, the distance between current data point x_i and all n existing cluster centres Cc_j are calculated as $D_{ij} = \|x_i - Cc_j\|$ where $j = 1, 2, \dots, n$ is the cluster index.

Step 2: The distance D_{ij} is calculated and compared with existing clusters' radius R_j . If at least for one R_j the condition $D_{ij} \leq R_j$ is satisfied, then x_i belongs to a cluster C_m with the minimum distance $D_{im} = \|x_i - Cc_m\| = \min(\|x_i - Cc_j\|)$ for $D_{ij} \leq R_j$ ($j = 1, 2, \dots, n$). In this case, neither a new cluster is created, nor will an existing cluster be updated (the cases x_4 and x_6 in Fig. 6.1(b) and (c)), and the algorithm returns to Step 1. Else, if $D_{ij} > R_j$, the algorithm proceeds to the next step.

Figure 5.1: Schematic of ECM clustering procedure using with samples x_1 to x_9 in 2-D space. (a) The first input, x_1 is adopted as the first cluster C_1^0 . (b) x_2 : updates cluster C_1^0 to C_1^1 ; x_3 : creates a new cluster C_2^0 ; x_4 : belongs to cluster C_1^1 thus no change is required. (c) x_5 : updates cluster C_1^1 to C_1^2 ; x_6 : belongs to cluster C_1^2 thus no change is required; x_7 : updates cluster C_2^0 to C_2^1 ; x_8 : creates a new cluster C_3^0 . (d) x_9 : updates cluster C_1^2 to C_1^3 (Figure adapted from Kasabov and Song, 2002).

Step 3: For all existing clusters, $S_{ij} = D_{ij} + R_j$ ($j = 1, 2, \dots, n$) is calculated. The cluster C_a with the minimum value of $S_{ia} = D_{ia} + R_a = \min (S_{ij})$ where $j = 1, 2, \dots, n$ is identified.

Step 3(i): If $S_{ia} > 2 \times D_{thr}$, the data point x_i does not belong to any existing clusters. Thus, a new cluster is created as described in Step 0 (the cases of x_3 and x_8 in Fig. 6.1) and the algorithm returns to Step 1. Else, the algorithm proceeds to the next step.

Step 3(ii): The cluster C_a with $S_{ia} \leq 2 \times D_{thr}$ is updated by increasing its radius R_a to $R_a^{new} = S_{ia} / 2$. The new cluster centre Cc_a^{new} is located at the point on the line connecting x_i and Cc_a , and the distance to the point x_i is made equal to R_a^{new} (the cases x_2, x_5, x_7 , and x_9 in Fig. 6.1). The algorithm then returns to Step 1.

Figure 6.2: Schematic illustration of ECM algorithm.

6.2.2 Learning process in DENFIS

DENFIS uses the Takagi-Sugeno (Takagi and Sugeno, 1985) type fuzzy inference system where the i^{th} rule is:

$$\text{If } x_1 \text{ is } R_{i1} \text{ and } x_2 \text{ is } R_{i2} \text{ and } \dots \text{ and } x_q \text{ is } R_{iq}, \text{ then } y \text{ is } f_i(x_1, x_2, \dots, x_q) \quad (6.2)$$

and y is the consequent, “ x_j is R_{ij} ” ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, q$) are $m \times q$ fuzzy propositions as m antecedents form m fuzzy rules, q is the number of inputs used, x_j ($j = 1, 2, \dots, q$) are antecedent variables defined over the universes of discourse, X_j ($j = 1, 2, \dots, q$), R_{ij} ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, q$) are fuzzy sets defined by their fuzzy membership functions given by $\mu_{R_{ij}} : X_{ij} \rightarrow [0,1]$. The membership function used is the Gaussian membership function given by:

$$\mu(x) = MF(x, \sigma, c) = \exp\left(\frac{-(x-c)^2}{2\sigma^2}\right) \quad (6.3)$$

where c is the cluster centre; σ is the width and is usually defined as a multiple of D_{thr} , where D_{thr} is the threshold (clustering parameter) value which controls the size and hence the number of clusters to be created. In the present study $\sigma = 0.989 \times D_{thr}$, as recommended by Kasabov and Song (2002) was used. The number of membership functions is found automatically during the training process. DENFIS adopts the first-order Takagi-Sugeno fuzzy inference systems where $f_i(x_1, x_2, \dots, x_q)$, $i = 1, 2, \dots, m$, are linear functions. The equation for the consequent part is given by:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_q x_q \quad (6.4)$$

where $\beta_0, \beta_1, \beta_2, \dots$, and β_q are parameters to be optimised during the learning process.

In DENFIS, the first n_0 data pairs where $n_0 = q + 6$ (Kasabov and Song, 2002) are used to create the first m (m is the number of initial rules) cluster centres by the evolving clustering method. For each cluster centre C_i ($i = 1, 2, \dots, m$) the most p_i nearest data points to the centre are found. Using Eqn. (6.3) and the position of the centre, the antecedents of the rule corresponding to that specific cluster will be found. The above-mentioned p_i data points and their distances to the cluster centre C_i are then used to estimate the consequent parameters m , n_0 , and p_i .

When a new data pair is presented to DENFIS, a new fuzzy rule will be created if ECM creates a new cluster centre ($S_{ij} > 2 \times D_{thr}$) for this data point (Fig. 6.2). Using the position of this new cluster centre and Eqn. (6.3), the antecedent of the new rule will be formed. In order to estimate an initial consequence function for the newly established fuzzy rule, an existing fuzzy rule will be found where the rule node is closest to the newly established rule node. On the other hand, if the distance between the new data pair and a rule node (S_{ij}) is less than $2 \times D_{thr}$ then that existing fuzzy rule will be updated using weighted recursive least square estimator (LSE) method (Hsia, 1977; Goodwin and Sin, 1984) with forgetting factor which takes values typically between 0.8 to 1 (Kasabov and Song, 2002). In addition, ECM may change the rule node position as mentioned in step 3(ii) in Section 6.2.1 in which case, the corresponding rule will be updated through changing its antecedents.

Choosing a small D_{thr} values may cause many unneeded clusters which in turns can lead to over-fitting. Over-fitting can be detected when the performance of the model during training increases while the performance in simulating the testing data set is decreasing. To avoid this, it is needed to conduct a sensitivity analysis by training/verification data sets to find the suitable value of D_{thr} which gives high performance in both training and verification phases.

6.3 Methodology

5.3.1 Study sites and data used

Rainfall and runoff data collected from three different catchments ranging from a small experimental catchment (Catchment 1), a small urbanised catchment (Catchment 2), and a large rural watershed (Catchment 3) were analysed for this study (see Section 3.5). Data from Catchment 1 comprises 10 storm events obtained for the period from October 2002 to December 2002 (Appendix A). Among the 66 events available for Catchment 2 (Appendix B), the discharge and rainfall measurements for 63 events recorded at 5 min intervals at the outlet of the catchment were used in this study since the remaining 3 events (12 Mar 05, 27 Mar 05, 21 Jan 06) were considered to be atypical. There was no correlation between

rainfall and runoff time series in these three events. The data for Catchment 3 comprises the continuous rainfall and discharge recorded from 1 January 1990 to 21 January 2006.

6.3.2 Physically-based models used

In this study, the results of DENFIS model for Catchments 1, 2, and 3 were compared with physically-based models Kinematic Wave Model (KWM), Storm Water Management Model (SWMM), and University Regina Hydrologic Model (URHM), respectively (see Section 3.3).

6.3.3 Input data selection and model development

DENFIS model input consisted of antecedent rainfall for Catchment 1 and Catchment 2 while for Catchment 3 antecedent rainfall and discharge were used. The model output for all catchments was runoff at the current time $Q(t)$. Rainfall-runoff modelling can be carried out either on a continuous basis or event-based. Catchments 1 and 2 are smaller in size and event-based analysis is pertinent. In event-based analysis, the event used for training the model is an important factor determining model performance. Following the procedure discussed for training ANFIS in Catchment 1 (Section 4), same events were chosen as candidates for training. Then by employing MICA input selection method (Section 3.9.3) the three high ranked non-sequential rainfall input combinations were chosen. A cross-validation analysis was carried out to find the best training event and input combination. It was found that Event 27 Oct 2002 was the best event to be chosen as the training event for Catchment 1. The best input combination found for Catchment 1 is provided in Table 6.1. Adopting the same training events of model ANFIS4 (Section 5.4) and using MICA input selection method, three high ranked input combinations were found. Considering one verification event (25 May 2005) the best input combination for Catchment 2 was found. Details of the events and inputs used for training the DENFIS model for Catchments 1 and 2 are shown in Table 6.1.

Table 6.1: Events and inputs used for DENFIS training in Catchments 1 and 2.

Catchment	Training Events	Verification Events	Inputs
1	27 Oct 2002	13 Nov 2002	R(t-3),R(t-8)
		18 Nov 2002	
		22 Nov 2002	
	19 Jul 2005	17 Sep 2006	R(t-3),R(t-7),R(t-12)
2	30 Jul 2005	04 Jul 2005	
	24 Dec 2005	16 May 2005	
		06 Apr 2006	

Rainfall-runoff modelling in Catchment 3 is based on a continuous time series. From approximately 16 years of available rainfall-runoff time series, 10 years (1 Jan 1996 to 31 Dec 2005) data was used for training, 1 year (1 Jan 1995 to 31 Dec 1995) for verification and 5 years (1 Jan 1990 to 31 Dec 1994) for testing. In the selection of training, verification, and the testing data, care was taken to ensure that extreme events were fairly distributed in all these three data sets. The three input selection methods discussed in Section 3.9 were used to find the input combination candidates. These analyses were firstly done considering rainfall antecedents only. Results obtained through verification stage revealed that using rainfall inputs only was not able to capture the rainfall-runoff process in this catchment since performance evaluations statistics such as CE and r^2 were smaller than 0.5 (not satisfactory). This is attributed to the fact that there is only one rain gage in this large watershed and its location is very close to the outlet of the catchment which makes the rainfall data not truly representative of the rainfall for the entire watershed. Therefore, it was decided to consider discharge antecedents in input selection procedure. The verification results showed that the input combination which included two discharge antecedents as inputs performed well. Details of the inputs found for Catchment 3 during the verification phase and used for testing phase are provided in Table 6.2. It is worth mentioning that the same input selection procedure for ANFIS was resulted in the use of $R(t)$, $R(t-1)$, $R(t-2)$, and $Q(t-3)$ as inputs.

Table 6.2: Details of data and inputs used for DENFIS training in Catchment 3.

	Training Data	Verification Data	Testing Data	Inputs
Number of Days	3,674	365	1826	R(t-1), R(t-5), Q(t-1),Q(t-3)
Starting Date	01 Jan 1996	01 Jan 1995	01 Jan 1990	
Closing Date	21 Jan 2006	31 Dec 1995	31 Dec 1994	

The parameters that are required as inputs by the user are the threshold (D_{thr}), n_0 , and m (see Section 6.2). All other parameters including the number of membership functions are optimized during model learning. D_{thr} is a clustering parameter which determines the size of the clusters in the ECM algorithm and hence the number of rules that will be determined by the model. A sensitivity analysis varying D_{thr} over the range 0.05 to 0.2 was carried out to test the sensitivity of D_{thr} for the study datasets. It was found that $D_{thr} = 0.1$ gave the best results for all the three catchments. It is worth mentioning that 0.1 is also the recommended value for D_{thr} , as mentioned in Kasabov and Song (2002). The initial value of m (smallest number of initial rules to be created) adopted was 3 as this was found to give the best results. The final number of the rules is dependent on the learning process. In the original paper on DENFIS algorithm (Kasabov and Song, 2002) the authors used triangular membership functions. However, in the present study some trials were performed with both triangular and Gaussian membership functions. The results revealed that using Gaussian membership functions gave better performance compared to triangular one. Therefore, Gaussian membership functions were adopted for all catchments of this study. Moreover, the stopping criterion of the model as recommended by Kasabov and Song (2002) was set to 0.0001.

6.4 Results and Discussions

DENFIS model results were compared with the results obtained by KWM for Catchment 1. A further simulation was also run with ANFIS for comparison between online and offline models. In the KW model, ϕ -index of 2.7 mm/hr corresponding to the loss rate for Event 6 was adopted in the simulation of the 10

testing events. This value of the ϕ -index was adopted as it was the median ϕ -index value for the 10 events (Chua et al., 2008). Figures 6.3(a) – (e) show the CE , r^2 , $RMSE$, MAE and RPE values, respectively, for the testing events obtained by the DENFIS model and compared with the results obtained by the ANFIS and KW models. It can be seen that DENFIS results are comparable with the results obtained by the KWM and ANFIS models. As the experimental catchment is a completely impervious rectangular plane surface, the KWM is expected to model the events well since all the model parameters, except the loss rate, are known accurately. Thus, the good agreement between DENFIS and KWM results validates the performance of the DENFIS model for Catchment 1. The hydrographs predicted for the 10 events (note that Event 3 is the training event) are shown in Fig. 6.4 and compared with the measured and KWM simulated hydrographs. It can be observed that the hydrographs simulated by the DENFIS model match the measured hydrographs well.

Figure 6.3: Comparison between different models used for Catchment 1. (a) CE , (b) r^2 , (c) $RMSE$, (d) MAE , and (e) RPE .

Ev1: 06 Oct 2002

Ev2: 14 Oct 2002

Ev3: 27 Oct 2002

Ev4: 03 Nov 2002

Ev5: 13 Nov 2002

Ev6: 17 Nov 2002

Ev7: 18 Nov 2002

Ev8: 22 Nov 2002

Figure 6.4: Hyetograph, observed and simulated hydrograph for events simulated by DENFIS and KWM in Catchment 1.

For further comparison the average values of CE , r^2 , $RMSE$, MAE and RPE values over all events obtained by DENFIS, ANFIS, KWM, and an ARX model (The same ARX model presented in Section 4.7) are compared in Table 6.3. In general, DENFIS performs comparable to ANFIS and KWM models while it slightly outperforms ARX model.

Table 6.3: Comparison of DENFIS, ANFIS, KWM, and ARX models for Catchment 1.

Model	CE	r^2	$RMSE$ (l/s)	MAE (l/s)	RPE
DENFIS	0.948	0.954	0.036	0.022	0.071
ANFIS	0.950	0.957	0.034	0.022	0.060
KWM	0.952	0.965	0.030	0.018	0.089
ARX(0,6,3)	0.944	0.952	0.040	0.023	0.090

DENFIS model results were compared with the results obtained by SWMM (SWMM1 in Section 5) based on the study by Tan et al., (2008) for Catchment 2. Figures 6.5(a) – (e) show the box plots of CE , r^2 , $RMSE$, MAE and RPE for the 63 events simulated by DENFIS and SWMM. Based on this analysis, it is shown that

DENFIS model results are comparable to SWMM and almost identical to results obtained by ANFIS. The median *RPE* obtained by DENFIS and ANFIS models are 0.241 and 0.238, respectively and are better compared with SWMM which has a median *RPE* value of 0.357.

Figure 6.5: Comparison between different models used for Catchment 2. (a) CE , (b) r^2 , (c) $RMSE$, (d) MAE , and (e) RPE .

Qualitative assessment of the predicted hydrographs was carried out by comparing the predicted and measured hydrographs of a representative event for each of the models in Figs. 6.6(a) and 6.6(b). Events 26 Jul 2005 was chosen as representative of the DENFIS model while the 16 Apr 2005 event was chosen as representative of the SWMM model. These are events which produced the median $RMSE$ for the individual models. Figures 6.6(a) and 6.6(b) show that in general, the DENFIS model is able to reproduce the hydrograph shape reasonably well. However, there are indications of fluctuations in the recessing limb of the hydrographs produced by the DENFIS model.

The average values of CE , r^2 , $RMSE$, MAE and RPE for the 63 events simulated by DENFIS, ANFIS, SWMM, and ARX models are compared in Table 6.4. It should be noted that the ANFIS, SWMM, and ARX models in Table 6.4 are same as those were discussed in Section 5.3.3. As can be seen, results obtained by DENFIS are comparable with ANFIS and SWMM models. However, both DENFIS and ANFIS outperform SWMM model in peak estimation. Moreover, DENFIS results are superior to ARX(0, 6, 4) model for all the error statistics.

Figure 6.6: Hyetograph, observed and simulated hydrograph for median events simulated by DENFIS and SWMM for Catchment 2.

Table 6.4: Comparison of DENFIS, ANFIS, SWMM, and ARX models for Catchment 2.

Model	CE	r^2	$RMSE$ (m^3/s)	MAE (m^3/s)	RPE
DENFIS	0.607	0.716	2.864	1.727	0.242
ANFIS	0.570	0.720	2.970	1.933	0.229
SWMM	0.642	0.770	2.648	1.719	0.348
ARX(0,6,4)	0.423	0.595	3.248	2.158	0.358

The results of the DENFIS model are compared with results obtained from the URHM model, ANFIS, and ARX model for Catchment 3 in Table 6.5. ARX model calibrated following the approach discussed in Section 4.7. As can be seen, DENFIS gives slightly better performance when compared with URHM model since the CE and r^2 values are higher and $RMSE$ and MAE are lower, compared to URHM model results. Similar to Catchments 1 and 2, DENFIS and ANFIS showed almost identical performance. DENFIS outperforms ARX model. The distribution of RPE determined for 27 peak discharges where $Q_{\text{observed}} > 100 \text{ m}^3/\text{s}$ is shown in

Fig. 6.7. DENFIS performs best in peak discharge estimation (average = 0.137, median = 0.144) when compared with URHM (average = 0.296, median = 0.277) and ANFIS (average = 0.255, median = 0.272) models. Moreover the *RPE* for DENFIS spreads over a smaller range, [0.015, 0.261] compared to [0.022, 0.691] for URHM and [0.005, 0.577] for ANFIS. Significantly, this indicates that the local learning algorithm is able to provide better estimates of peak discharge compared to batch learning (ANFIS).

Table 6.5: Comparison of *CE*, r^2 , *MAE*, and *RMSE* values for DENFIS, ANFIS, and URHM models for Catchment 3.

Model	<i>CE</i>	r^2	<i>RMSE</i> (m ³ /s)	<i>MAE</i> (m ³ /s)
DENFIS	0.821	0.823	18.091	7.400
ANFIS	0.840	0.850	17.133	6.550
URHM	0.758	0.760	21.052	11.189
ARX(3,3,0)	0.736	0.736	21.962	8.992

Figure 6.7: Comparison of *RPE* values for peak discharges $> 100 \text{ m}^3/\text{s}$ for Catchment 3.

Figure 6.8 shows the observed rainfall and discharge time series for Catchment 3 and is compared with the discharge time series simulated by DNFIS and URHM models. As shown, the DNFIS model is able to predict both the high and low flows well and the predicted hydrograph is comparable to the URHM model result. A notable exception is the under prediction of the highest discharge recorded in the testing dataset (time = 1,629 days). This highlights the problem of extrapolation with data driven methods, in general, since this class of models do not work as well if the peak discharge to be simulated far exceeds the peak discharge used in training the model. It was possible to include the highest discharge value in the dataset used to train the DNFIS model (which would have resulted in an improvement in results), but this was avoided since the intention was to use the same dataset for training both the DNFIS and URHM models, in order to facilitate a more realistic comparison between the models. Nevertheless, the *RPE* obtained by DNFIS for this event was 0.248, which is not much worse than the URHM model where an *RPE* of 0.178 obtained. This highlights the general difficulty in making accurate predictions of extreme events faced by all models. Lastly, it is worth noting that the DNFIS model used discharge antecedents in its inputs. This was inevitable since the rainfall recorded at Beimen (Fig. 3.16) was not truly representative of the rainfall for the entire watershed as it is situated close to the outlet of the watershed. In fact, the URHM model utilised additional (August 1, 2004 to January 21, 2006) rainfall data obtained from independent sites in its calibration (Qin et al., 2010) which was not made available for our study. This highlights an important distinction between the DNFIS and URHM models. The URHM model requires accurate rainfall inputs since precipitation is the primary driver for the model; DNFIS on the other hand does not need complete precipitation information but the trade-off is that measured discharge is required.

Figure 6.8: Measured rainfall, observed discharge, and simulated discharges by DENFIS and URHM models for Catchment 3.

As seen in the results for all three catchments, DENFIS and ANFIS showed almost similar performance in terms of the error statistics considered. In terms of training times, however, DENFIS is faster than ANFIS. This is consistent with observations by Hong and White (2009). Table 6.6 compares the training process in DENFIS and ANFIS for Catchment 3. As a consequent of the online training algorithm, more rules were created in DENFIS (30 rules) and took 10 seconds. This is compared with the 16 rules created by ANFIS, which utilises batch learning, but required a longer time (31 seconds). This reduction in training times is inherent in the online learning process since it utilises a one pass algorithm, compared to the batch learning approach adopted by ANFIS which requires 60 epochs for training with the same dataset. Specifically, during batch learning, the input space is initially partitioned randomly and then during a specific number of epochs will converge to the final rule nodes based on a global learning process. In DENFIS however, the number of rules are gradually updated as each data point is introduced to the one pass training algorithm.

Table 6.6: Evaluating the training process in DENFIS and ANFIS.

Model	No. of Rules	Training Time (sec)	Epochs	CE	r^2	$RMSE$	MAE
DENFIS	30	10	1	0.765	0.770	17.071	7.547
ANFIS	16	31	60	0.854	0.854	13.457	6.745

Figure 6.9 demonstrates the rule creation procedure with the online learning algorithm in DENFIS for Catchment 3. Although the inputs consist of 4 parameters, $R(t-1)$, $R(t-5)$, $Q(t-1)$, and $Q(t-3)$, for illustrative purposes, only the time series of $Q(t-1)$ is plotted since this is expected to be the dominant input parameter. It can be observed that the number of rules increase whenever a spike in the $Q(t-1)$ occurs, showing the response of the model to significant changes in the time series.

Figure 6.9: Rule creation as a function of input $Q(t-1)$.

Table 6.7 shows the evolution of the rule creation procedure in DENFIS. As shown in the table, the first cluster is created with the first data point, it has null radius. Based on the value of D_{thr} specified, the second cluster centre is created only after the 8th data point. At this time, a new cluster is formed and the initial cluster has grown in size due to the addition of new data points into the cluster. As mentioned in Section 6.2, in DENFIS the consequent function is created when two conditions are satisfied: passing at least 10 data samples ($n_0 = q + 6$) and a minimum 3 rules ($m = 3$) created. These conditions are satisfied, after passing 119 data samples, the consequent linear function is created. With the input of the remaining data samples, more clusters were created and the existing rules updated. The positions of the training data on the $Q(t-1)$ time series corresponding to the data sequence mentioned in Table 6.7 are indicated by arrows in Fig. 6.9. The last cluster centre is produced by 3,801st data sample. By the last data sample (4,039th) all the 30 existing rules and their parameters including cluster centres position, radiuses, and parameters of the linear function of consequent part are updated to their final values.

Table 6.7: Creating and updating the rules in DENFIS model for normalized data in Catchment 3.

Sequence of Training Data	Rule Number	Cluster Centre				Radius	Consequent Linear Function $aX_1+bX_2+cX_3+dX_4+e=0$				
		X_1	X_2	X_3	X_4		a	b	c	d	e
1	1	0.115	0.045	0.045	0.045	0	-	-	-	-	-
8	1	0.171	0.052	0.091	0.061	0.088	-	-	-	-	-
	2	0.045	0.338	0.104	0.119	0	-	-	-	-	-
119	1	0.171	0.052	0.091	0.061	0.088	0.0105	0.1607	-0.0010	0.8097	-0.0405
	2	0.046	0.186	0.102	0.114	0.076	0.0106	0.1346	0.0043	0.8040	-0.0227
	3	0.433	0.052	0.104	0.109	0	0.0909	0.0641	0.3886	0.7005	-0.8366
558	1	0.171	0.052	0.091	0.061	0.088	0.0103	0.0980	0.0059	0.8310	-0.0680
	2	0.046	0.178	0.107	0.113	0.081	0.0105	0.0324	0.0055	0.8603	-0.0508
	3	0.314	0.066	0.114	0.157	0.1	-0.0074	0.1180	-0.0540	0.3951	0.5606
	4	0.074	0.342	0.205	0.126	0.1	0.0052	0.2577	-0.0305	0.3582	0.4288
	-										
1269	9	0.504	0.045	0.063	0.063	0	-0.0072	0.1167	-0.0540	0.3956	0.5608
	1	0.171	0.052	0.091	0.061	0.088	-0.1476	1.0248	0.6191	1.5209	-0.1394
	2	0.055	0.179	0.109	0.118	0.086	0.0124	0.4800	-0.0082	0.5014	0.1006
	-										
3801	17	0.093	0.049	0.558	0.547	0	-0.0193	0.9455	-0.0807	0.5954	0.0403
	1	0.176	0.052	0.103	0.062	0.095	-0.0085	0.6086	0.0339	0.4334	0.0714
	2	0.055	0.179	0.109	0.118	0.086	0.0381	0.0936	0.0020	0.3971	-0.0106
	-										
	29	0.175	0.226	0.162	0.254	0.077	-0.0494	0.5577	0.1032	0.6765	0.0922
4039	30	0.047	0.299	0.428	0.125	0	-0.0078	0.2550	0.0254	0.7451	0.0180
	1	0.176	0.052	0.103	0.062	0.095	0.0194	0.0773	-0.1742	0.6764	0.1046
	2	0.055	0.179	0.109	0.118	0.086	0.0188	0.0801	-0.1658	0.6754	0.1054
	-										
	29	0.175	0.226	0.162	0.254	0.077	-0.0658	0.5972	0.1265	0.6737	0.1354
4039	30	0.142	0.271	0.410	0.162	0.054	-0.0162	0.4282	-0.0208	0.7661	0.0768

Figure 6.10 clarifies the rule creation procedure by making reference to the actual values of the inputs at the time of rule creation. The first rule is created with the first data point, during low flow conditions and the inputs are relatively small in magnitude. The second rule is created with the 8th data point of the training dataset when $R(t-5)$ is large. This data point occupies a location where $S_{ij} > 2D_{thr}$ for existing clusters in the input space and hence a new rule node is created. As can be observed in Fig. 6.10, the model responds to changes in the magnitude of the inputs, creating a new rule whenever the inputs fall into areas of the input space that are not included in existing clusters.

One of the major differences between a batch (global) learning process and an online (local) learning process is the number of rules needed to capture the input-output relationship. In batch training, the number of rules is pre-defined by the user since the number of membership functions have to be specified by the user. Thus, if n membership functions are specified for each of the input variables, n^q rules where q is the number of input variables will be produced. In most batch models, n has to be selected by trial and error. Although the number of rules can be increased by the specification of greater number of membership functions, this can result in over-fitting leading to deterioration in the model results. During local learning, the number of rules is added incrementally, and the final number of rules is dependent on the variety of combinations between input variables in the q -dimensional space. This results in an optimal allocation of rules corresponding to specific patterns in the input data stream. For Catchment 3, the total number of rules created by the DENFIS model was 30 which is almost twice that for ANFIS. This is due to the finer partitioning in input-output space corresponding to the relative distribution of the inputs. Although the higher number of rules in DENFIS compared to ANFIS didn't result to a better performance in DENFIS (both models gave almost identical results for all three catchments), DENFIS could be considered as a better model in this regards comparing to ANFIS, since there is no need to find the suitable number of membership functions by trial.

Figure 6.10: Combination between 4 different input variables for data points which create a new rule.

6.5 Effect of Data Length on DENFIS Model Training

For online learning, the learning process gradually progresses with each input of the training data while in batch learning, the whole training data set is used during training. Due to the difference between these two methodologies, the amount of data needed to effectively train a model is expected to be different. Usually, the data length required to adequately train an offline model is not known beforehand and has to be ascertained by trial and error. On the other hand, the data length required to effectively train an online model can be more easily ascertained since training proceeds incrementally. Therefore, a sensitivity analysis was carried out to address this issue using the data for Catchment 3. In order to avoid the influence of data quality on training, among the 11 years of training data used for Catchment 3 (Section 6.4), only five of the wetter years (1997, 1998, 1999, 2000, and 2002) with almost similar high peak discharges were selected for training. Figure 6.11 shows the rainfall and discharge time series for these 5 years. The discarded years belong to drier years with lower flows. The inputs and testing data set used in this analysis are similar to that used in Section 6.4.

The sensitivity analysis proceeded by increasing the training data length gradually from 1 year to a total of 4 years' data, in 1 year increments. Since different learning algorithms are used in DENFIS and ANFIS models, each data set may have different information for the training of these two models. For a fair comparison between DENFIS and ANFIS models, the 'best' training data for the respective models with lengths of 1, 2, 3, and 4-years was first found by cross-validation. Table 6.8 shows the 1, 2, 3, 4-years training dataset for training ANFIS and DENFIS.

Figure 6.11: Rainfall and discharge time series for the years: (a) 1997, (b) 1998, (c) 1999, (d) 2000, and (e) 2002.

Table 6.8: The best 1, 2, 3, 4-years training data for ANFIS and DENFIS.

Training Data Length	ANFIS	DENFIS
1 Year	1997	1997
2 Years	1998, 2002	1997, 2000
3 Years	1997, 1998, 2002	1997, 1999, 2000
4 Years	1997, 1998, 1999, 2002	1997, 1998, 1999, 2000

The CE and r^2 values calculated for the training and testing dataset for various data lengths of the training dataset are compared for the ANFIS and DENFIS models in Figs. 6.12(a) and (b), respectively. Figure 6.12(b) shows that DENFIS is able to achieve significantly better results with just 1 year of training data as the CE value attained a value of 0.802 with just 1 year of training data used and increased to a maximum of 0.823. ANFIS, on the other hand trained poorly with just 1 year of training data. By increasing the training data length, the CE value for the ANFIS model increased to a maximum value of 0.767, which is lower than the value of 0.823 achieved by DENFIS. The same trend is observed for the r^2 values. The observed and simulated time series by ANFIS and DENFIS with training data length of 1, 2, 3, and 4 years are shown in Figs. 6.13(a) – (d), respectively.

The results of this analysis suggest that models based on batch learning are more sensitive to the training data length compared to models implementing online learning. When only 1 year of training data is used, although the ANFIS model trained well, it performed poorly during the testing phase. The DENFIS model on the other hand performed well with limited training data. This suggests that local learning models have significant advantages when training data is limited. The ANFIS model requires at least 2 years of training data. It is also observed that ANFIS model testing results are consistently poorer than training results; however, although the training results are poorer for DENFIS compared to ANFIS, DENFIS testing results are superior to ANFIS test results. These differences can be attributed to the inherent behaviour of local learning in which the model is presented with

training data incrementally. As a result, the model is selectively updated. This means that the model is updated only when necessary, reacting only when there are significant changes to the patterns in the input training data time series. Batch learning on the other hand is insensitive to local changes and thus less responsive to changes in the training data.

Figure 6.12: Training and testing CE and r^2 values for different training data length:

(a) ANFIS and (b) DENFIS models.

Figure 6.13: Measured rainfall, observed and simulated discharges by ANFIS and DENFIS models for Catchment 3 trained using training data length of (a) 1-year, (b) 2-years, (c) 3-years, and (d) 4-years.

(b)

Figure 6.13 (Cont'd): Measured rainfall, observed and simulated discharges by ANFIS and DENFIS models for Catchment 3 trained using training data length of (a) 1-year, (b) 2-years, (c) 3-years, and (d) 4-years.

(c)

Figure 6.13 (Cont'd): Measured rainfall, observed and simulated discharges by ANFIS and DENFIS models for Catchment 3 trained using training data length of (a) 1-year, (b) 2-years, (c) 3-years, and (d) 4-years.

(d)

Figure 6.13 (Cont'd): Measured rainfall, observed and simulated discharges by ANFIS and DENFIS models for Catchment 3 trained using training data length of (a) 1-year, (b) 2-years, (c) 3-years, and (d) 4-years.

6.6 Effect of Data Sequence when Training Online Models

Learning and the corresponding rule creation in online models evolve gradually by passing the training data points in sequence. Intuitively, it can be expected that when historical data are used, the sequence in which the training data is presented to the model can have an influence on the online learning process. This can be distinguished from batch learning where the entire training dataset is presented and thus sequencing of input data will not have an influence.

An analysis was carried out to compare the possible effects of training data sequencing between offline and online models, represented by ANFIS and DENFIS, respectively. The rainfall-runoff data from Catchment 3 for the years 1996, 1999, and 1998 were chosen as representative of years with low (L), medium (M), and high (H) flows, respectively, were chosen for training. The rainfall and runoff time series for the selected data are shown in Fig. 6.14. As a result, six different combinations of training data set were defined: LMH, LHM, MLH, MHL, HLM, and HML. ANFIS and DENFIS models trained with each of these training data sets were used to simulate runoff time series for the period of 1990-1994, identical to the testing data set used in Section 6.4.

The observed and simulated discharge time series for ANFIS with DENFIS models trained by LMH, LHM, MLH, MHL, HLM, and HML training datasets are shown in Fig. 6.15(a) - (f), respectively. It can be observed that the hydrograph simulated by the model trained with the HML dataset (Fig. 6.15(f)) shows relatively better agreement at the higher flow rates. Among the 17 rules created using the HML training dataset, 14 were made by the first part of the data (year 1999 as representative of high (H) flows) which potentially could allocate more rules to high peak flows. In contrast, for LMH, only 8 rules were created by data points related to H portion. Since the testing dataset comprises several high peak flows, the better performance obtained when the model was trained using the HML dataset can be attributed to the allocation of a greater number rules to the high peak flows regime compared to the other cases.

Figure 6.14: Rainfall and discharge time series for years with (a) 1996 - low (L), (b) 1999 - medium (M), and (c) 1998 - high (H) flows.

Figure 6.15: Measured rainfall, observed discharge, and simulated discharge for ANFIS compared with DENFIS models trained with (a) LMH, (b) LHM, (c) MLH, (d) MHL, (e) HLM, and (f) HML training datasets for Catchment 3.

Figure 6.15 (Cont'd): Measured rainfall, observed discharge, and simulated discharge for ANFIS compared with DENFIS models trained with (a) LMH, (b) LHM, (c) MLH, (d) MHL, (e) HLM, and (f) HML training datasets for Catchment 3.

Figure 6.15 (Cont'd): Measured rainfall, observed discharge, and simulated discharge for ANFIS compared with DENFIS models trained with (a) LMH, (b) LHM, (c) MLH, (d) MHL, (e) HLM, and (f) HML training datasets for Catchment 3.

Figure 6.15 (Cont'd): Measured rainfall, observed discharge, and simulated discharge for ANFIS compared with DENFIS models trained with (a) LMH, (b) LHM, (c) MLH, (d) MHL, (e) HLM, and (f) HML training datasets for Catchment 3.

Figure 6.15 (Cont'd): Measured rainfall, observed discharge, and simulated discharge for ANFIS compared with DENFIS models trained with (a) LMH, (b) LHM, (c) MLH, (d) MHL, (e) HLM, and (f) HML training datasets for Catchment 3.

Figure 6.15: Measured rainfall, observed discharge, and simulated discharge for ANFIS compared with DENFIS models trained with (a) LMH, (b) LHM, (c) MLH, (d) MHL, (e) HLM, and (f) HML training datasets for Catchment 3.

A quantitative comparison is made between the ANFIS and DENFIS models in Fig 6.16. The results show that the ANFIS model is insensitive to the sequencing of the training dataset which is to be expected of a batch learning model. The CE , r^2 , $RMSE$, and MAE values are constant for all the training datasets. However, CE , r^2 , $RMSE$, and MAE values for DENFIS trained on the different training datasets vary. Figure 6.16(a) - (d) show that the DENFIS model trained on the HML training dataset performed the best among the DENFIS models. Table 6.9 shows the number of rules created by each of the different DENFIS models in comparison with ANFIS. The number of rules created by each of the DENFIS models is different, ranging from 16 to 19 rules. In ANFIS however, if n membership functions are specified for each of the input variables, n^q rules where q is the number of input variables will be produced. Therefore, when 2 membership functions are specified for each of the 4 inputs, $2^4 = 16$ rules are produced. Thus, the number of rules is fixed in ANFIS since the number of membership functions is pre-defined by the user (see Section 6.4). In DENFIS however, the number of rules can vary depending on the online rule creation process in online training.

Figure 6.16: CE , r^2 , $RMSE$, and MAE values obtained by ANFIS and DENFIS models trained with data with six different for sequences.

Table 6.9: Number of rules created in ANFIS and DENFIS models as a function of training data sequencing.

Sequence in Training Data	DENFIS	ANFIS
LMH	16	16
LHM	19	16
MLH	16	16
MHL	17	16
HLM	17	16
HML	17	16

6.7 Real Time DENFIS (RT-DENFIS)

6.7.1 Introduction

In both online and offline NFS (for e.g. DENFIS and ANFIS, respectively), once the model is trained, the structure of the FIS including its membership functions, position of rule nodes, and rule parameters are fixed during the testing phase. This is a common situation in real time systems, where the model parameters remain fixed during the testing or implementation phase. In such situations, the model is trained again only when its performance has deteriorated to some pre-determined level. One reason why such a state of affairs exists in today's real time systems is that most systems operate batch or offline models. For such models, the only recourse is to take the model offline and retrain the model. In fact, this is also true in systems that operate physically-based models where a re-calibration is necessary with more up to date information. With the implementation of online models such as DENFIS; however, the possibility for models to be continuously updated, even during the implementation phase can be realised. Such a model is termed Real Time DENFIS or RT-DENFIS. The advantages of such a system include:

- (i) Model training takes place continuously. Such a model can be termed a self-training or self-learning model. There is no need for the model to be taken offline, and minimum user intervention is required.
- (ii) RT-DENFIS is updated using the latest information. Such a system makes the model sensitive to current changes to the system. One of the most often heard criticisms of data driven models is its inability to respond to changes, for example brought on by changes in land use or climate, which consequently affects hydrological behaviour. A self-learning model will be able to overcome this deficiency since its model parameters are updated continuously.

The ECM algorithm shown in Fig. 6.2 was adopted in the development of RT-DENFIS, as shown in Fig. 6.17. For each new input data point the output will be simulated based on the available rules created from training with previous data. This phase is similar to the testing phase in DENFIS. The output of the model is

then fed back to the model to update the clusters and rules. This phase is similar to the learning phase in DENFIS. Hence, for the next incoming data, the model has been updated by the last data point. In other words there is no specific training data set in RT-DENFIS, but training is continuously in progress as each new data point is presented to the model. For example, in a discharge forecasting problem, RT-DENFIS can predict the one day ahead discharge $Q(t+1)$ by using the historical data till the present time t . One day later, the observed discharge at time $t + 1$ will be fed back to the model to update the clusters and rule nodes for predicting $Q(t + 2)$, and so on.

6.7.2 Application of RT-DENFIS in Discharge Forecasting

In order to compare the capabilities of RT-DENFIS with the commonly used training-testing procedure, 43 years rainfall, runoff and temperature time series from Catchment 4 were used since a long historical data is preferred to highlight the importance of model updating due to the possible changes in hydrologic behaviour. Based on the MICA input selection method (Section 3.9.3), $R(t)$, $R(t-1)$, $T(t)$, $T(t-1)$, and $Q(t)$ were chosen as inputs to predict $Q(t+1)$ as output. The following two scenarios were considered:

Scenario (1): The first year of data (1961) was used to train an ANFIS model but the model was used to forecast $Q(t+1)$ for the next year (1962) only. The model was subsequently retrained every year using all available data to date and used in the simulation phase to simulate $Q(t+1)$ over the following year period. Thus, training using data from 1961 - 1962 was used to simulate $Q(t+1)$ for 1963, and so on. This process was repeated until data for 1961 - 2002 was used for training and the model used to forecast $Q(t+1)$ for 2003. In this scenario, the performance of a batch model (ANFIS) updated once a year was investigated.

Figure 6.17: Development of RT-DENFIS model.

Scenario (2): Using the first year of data (1961) for initialization, an RT-DENFIS model was developed to simulate $Q(t+1)$ in real time mode from 1962 to 2003. In this scenario, RT-DENFIS uses the present day information to predict the discharge for the next day. At time t , the observed discharge $Q(t)$ will be used to update the model to predict the discharge for the following day $Q(t+1)$. This procedure is repeated to the last available data point. In this scenario, the effect of a local model (RT-DENFIS), trained automatically online can be investigated.

The *CE* and *MAE* values obtained by the ANFIS model for Scenario (1) and RT-DENFIS for Scenario (2) in simulating $Q(t+1)$ for the time period 1962 - 2003 are presented in Fig. 6.18(a) and (b), respectively. As can be seen, the ANFIS model trained and updated under Scenario (1) does not perform well for the first few years, up to about 1975. As shown in the figure, improvements are obtained especially for 1976 onwards. RT-DENFIS (Scenario (2)) shows significant improvements over ANFIS model, especially during the early simulation period, from 1962 – 1975. This indicates the strength of online models, where a shorter data length is required. This is a significant point, since one of the criticisms of batch learning models is the large amount of data required for training (Govindaraju, 2000). It is noticed in Fig. 6.18(a) that ANFIS performance deteriorates drastically in 1969. An investigation of the training data over the entire input space revealed that approximately 64% of the data points accounted for only 2 out of the 32 rules in total (2 out of 32 input space partitions) with the remaining 36% of the data distributed over the remaining rules. Therefore, the trained ANFIS model overestimate or underestimate the 31% of the testing data set (1961) which is not belonging to the partitions that ANFIS is trained for. This is consistent with the fact that data-driven models perform well for what they have been trained for.

The improvements of RT-DENFIS over ANFIS are modest for 1980 – 2003 and this is attributed to the fact that the ANFIS model has sufficient data from training and that the model benefits from retraining after every year. However, it should be mentioned that the results for RT-DENFIS were achieved without having the need to retrain the model as the model is self-training.

Figure 6.18: Performance of ANFIS model for Scenario (1) and RT-DENFIS for Scenario (2) in simulating $Q(t+1)$ in terms of: (a) CE and (b) MAE .

In practical situations, offline models are usually trained with an amount of data and the trained model will be used for simulating the remaining data. In order to compare the results of such model with ones from RT-DENFIS, an ANFIS model was developed based on a new scenario denoted as Scenario (3) in this study:

Scenario (3): The first ten years of data (1961 - 1970) was used to train an ANFIS model. Then the trained models were employed to simulate a one day ahead discharge $Q(t+1)$ for the remaining (1971 - 2003) period. In this scenario, the possible effects that may occur when a batch model becomes out-dated can be

investigated. The CE and MAE values obtained by the ANFIS model for Scenario (3) and RT-DENFIS for Scenario (2) in simulating $Q(t+1)$ for the time period 1971 - 2003 are presented in Fig. 6.19(a) and (b), respectively. As shown in Fig. 6.19, RT-DENFIS outperforms the ANFIS model for Scenario (3) in terms of CE and MAE . This indicates the fact that a trained model can become out-dated in time if it is not updated periodically and significantly shows the self-training capability of RT-DENFIS.

Figure 6.19: Performance of ANFIS model for Scenario (3) and RT-DENFIS for Scenario (2) in simulating $Q(t+1)$ in terms of: (a) CE and (b) MAE .

For further comparison, the CE , r^2 , $RMSE$, and MAE values obtained by the ANFIS, ARX, and Naïve models for Scenario (3) and RT-DENFIS for Scenario (2) in simulating $Q(t+1)$, $Q(t+2)$, and $Q(t+3)$ for the time period 1971 - 2003 are presented in Table 6.10. The ARX model which used $Q(t)$, $Q(t-1)$, $Q(t-2)$, $R(t)$, $R(t-1)$, $R(t-2)$, $R(t-3)$, $T(t)$, $T(t-1)$, and $T(t-2)$ as its inputs was found to be the best ARX model among all possible combination of inputs. As can be seen, RT-DENFIS consistently outperforms all other models in simulating $Q(t+1)$, $Q(t+2)$, and $Q(t+3)$. This supports the good potential of this model in forecasting problems.

Table 6.10: Comparison of CE , r^2 , $RMSE$, and MAE values for RT-DENFIS, ANFIS, ARX, and Naïve models in predicting $Q(t+1)$, $Q(t+2)$, and $Q(t+3)$.

Model	CE	r^2	$RMSE$ (m^3/s)	MAE (m^3/s)
$Q(t+1)$				
RT-DENFIS	0.923	0.924	0.299	0.157
ANFIS	0.908	0.917	0.325	0.180
ARX	0.752	0.761	0.596	0.300
Naïve	0.848	0.853	0.467	0.221
$Q(t+2)$				
RT-DENFIS	0.773	0.782	0.524	0.274
ANFIS	0.717	0.754	0.577	0.309
ARX	0.531	0.565	0.819	0.461
Naïve	0.596	0.637	0.760	0.360
$Q(t+3)$				
RT-DENFIS	0.584	0.610	0.711	0.385
ANFIS	0.545	0.591	0.738	0.423
ARX	0.385	0.447	0.937	0.563
Naïve	0.413	0.499	0.916	0.448

6.8 Conclusions

The following can be concluded from this study:

- (i) Dynamic Evolving Neural-Fuzzy Inference System (DENFIS) was tested against physically-based models for 3 catchments, with different sizes and different modelling complexities. Generally, DENFIS was comparable if not superior, in some respects, compared to the physically-based models used.
- (ii) Although the improvement in results obtained by the local learning algorithm over the global offline learning algorithm used in ANFIS were modest, DENFIS had a shorter training time compared to ANFIS. This is because the local learning uses a one pass algorithm, compared to global learning which requires several epochs for training.
- (iii) DENFIS could be considered as a better model in finding the proper number of membership functions comparing to ANFIS, since there is no need to find the suitable number of membership functions by trial.
- (iv) An analysis of the continuous time series for a large rural (Heshui) watershed indicates that the local learning algorithm has better capabilities in predicting the peak discharge compared to a model based on global learning.
- (v) DENFIS showed less sensitivity to the length of training data while ANFIS was found to be sensitive to the length of the training data used. Significantly, DENFIS required shorter length of training data compared to ANFIS. This represents a significant advantage especially in situations where data is limited.
- (vi) DENFIS was found to be sensitive to the sequence of training data while ANFIS was not. The use of a training dataset where the data was arranged in a sequence which comprised of high flows followed by medium and low flows gave better results.
- (vii) Batch models trained with historical data need to be retrained periodically in order to maintain reasonably good predictions. The real time dynamic

evolving neural fuzzy inference system (RT-DENFIS) however, does not need retraining since the model is constantly updated. This characteristic of RT-DENFIS could be advantageous in dealing with any possible temporal or spatial changes of the catchment as no specific retraining process is needed.

CHAPTER 7 SIMILARITY-BASED FUZZY NEURAL NETWORK FOR RAINFALL-RUNOFF MODELING

7.1 Introduction

To date several classes of NFS have been introduced in the literature. As mentioned in Section 3.2, FIS can be categorized as: (1) linguistic or Mamdani-type NFS (Mamdani and Assilian, 1975) and (2) precise or Takagi-Sugeno type NFS (Takagi and Sugeno, 1985). In addition, NFS can also be categorized based on the reasoning schema adopted. Four different reasoning schemas mentioned in the literature are: (1) Compositional Rule of Inference (CRI), (2) Truth Value Restriction (TVR) inference (Zadeh, 1975a, 1975b and 1975c), (3) Approximate Analogical Reasoning Schema (AARS) (Turksen and Zhong, 1990) and (4) Yager inference rule (Keller et al., 1992). CRI is the commonly used reasoning schema in rainfall-runoff modeling and is essentially based on matrix operations which are computationally simple. However, the effect of these matrix operations on the membership value propagation is not conceptually clear. As a result, there is the lack of a clear insight about CRI behavior (Turksen and Zhong, 1988). AARS is one of the approaches that avoid the pitfalls of CRI. This reasoning schema employs a similarity measure (SM) to determine whether a rule should be fired for a specific observation in the pattern matching phase. The same similarity measure is used to construct a modification function (ModF) to modify the right side of the rule in the consequent deduction phase.

In the present study, the details of an Approximate Analogical Reasoning Schema (AARS) are first presented. Next, an NFS model developed based on AARS, known as Pseudo Outer Product Fuzzy Neural Network (POP-FNN) (Quek and Zhou, 1999) will be tested for a rainfall-runoff modeling problem.

7.2 Approximate Analogical Reasoning Schema (AARS)

Given an observed fact A' and a simple fuzzy rule “if A then B ”; the basic procedure in AARS is to modify the consequence B of the fuzzy rule according to the

similarity between observed fact A' and the antecedent A . Being similar enough in comparison to a threshold, a rule can be fired and the conclusion B' can be deduced using a modification function which will be constructed based on the same similarity measure (Quek and Zhou, 1999). This process is shown schematically in Fig. 7.1. As can be seen, a rule will be fired to estimate the true proposition of the consequence (B) when the observed variable (A') is similar enough (SM bigger than a threshold) to a true proposition (A) of the antecedent. Then by using a modification function, the conclusion (B') will be deduced from the true proposition B .

Figure 7.1: Schematic process of approximate analogical reasoning schema in fuzzy inference.

7.2.1 Similarity Measures

The theoretical analysis of similarity relations has been dominated by geometric models. In these models, objects are represented as points in a space such that the observed dissimilarity among objects corresponds to the metric distance between the representative points. In derivation of similarity measures Zwick et al., (1987) concluded that similarity measures (SM) are actually distance measures (DM). This can be expressed by Eqn. (7.1):

$$SM = \frac{1}{1 + DM} \quad (7.1)$$

An identical definition has been also adopted in the study by Turksen and Zhong, (1990). In a behavioral experiment by Zwick et al., (1987) several different distance measures were generalized to fuzzy sets and a correlation analysis was done to assess their quality. Among the different measures studied, five were

reported to be suitable for distinguishing between the degree of similarity or dissimilarity. These are:

(1) Disconsistency Measure:

$$S_4(A, A') = 1 - \sup_{x \in X} \mu_{A \cap A'}(x) \quad (7.2)$$

where *sup* refers to supremum or the least upper bound of a set; $\mu_{A \cap A'}(x) = \min [\mu_A(x), \mu_{A'}(x)]$, for any $x \in X$.

(2) Hausdorff Measure (∞):

$$q_\infty(A, A') = \sup_{\alpha \geq 0} q(A_\alpha, A'_\alpha) \quad (7.3)$$

(3) Hausdorff Measure (*):

$$q_*(A, A') = q(A_{1.0}, A'_{1.0}) \quad (7.4)$$

where

$$q(U, V) = \max \left\{ \sup_{v \in V} \inf_{u \in U} d_2(u, v), \sup_{u \in U} \inf_{v \in V} d_2(u, v) \right\} \quad (7.5)$$

and is called the Hausdorff Distance in which *inf* stands for infimum or the greatest lower bound of a set and $d_2(U, V)$ is the Euclidian Distance. It is worth mentioning that the Hausdorff Measure (*) is a special instance of the Hausdorff Distance (∞).

(4) Kaufman and Gupta Measure (∞):

$$\Delta_\infty(A, A') = \sup_{\alpha \geq 0} \Delta(A_\alpha, A'_\alpha) \quad (7.6)$$

(5) Kaufman and Gupta Measure (*):

$$\Delta_*(A, A') = \Delta(A_{1,0}, A'_{1,0}) \quad (7.7)$$

where

$$\Delta(A_\alpha, B_\alpha) = \frac{(|a_1 - b_1| + |a_2 - b_2|)}{2(\beta_2 - \beta_1)} \quad (7.8)$$

and $[a_1, a_2]$, $[b_1, b_2]$ are the supports of A_α , B_α , respectively. $[\beta_1, \beta_2]$ is the support of both A_α , B_α , where $\alpha \in [0, 1]$. Also, Kaufman and Gupta (*) is a special instance of the Kaufman and Gupta Measure (∞).

7.2.2 Modification functions (ModF)

In AARS, a fuzzy rule “if A then B ” is to be fired by using a ModF which modifies the consequent B of the rule based on the SM between the observed fact A' and the antecedent A . ModF is dependent on the SM. In order to adjust the form of ModF in a way that the whole system can function as close as possible to the real situation, prior experience or historical data is required. It is worth mentioning that ModF can bypass the matrix operations of CRI for inferring the conclusion. Many forms of ModF could be defined but Turksen and Zhong (1988) introduced two types of ModF as follows:

(1.) Expansion form (more or less form):

The expansion form is equivalent to the linguistic ‘more or less’ and derives the conclusion B' by using the following equations:

$$B' = \min(1, \frac{B}{SM}) \quad \mu_{B'} = \min(1, \frac{\mu_B}{SM}) \quad (7.9)$$

where μ_B and $\mu_{B'}$ are the membership functions of the consequence B and the derived conclusion B' , respectively. Figure 7.2(a) illustrates the effect of the expansion form of the modification function. In this ModF, large values of distance measure increase the value of membership function (see Fig. 7.2(a)) and an

additional uncertainty will be introduced in such a form of the ModF. In an extreme case when the observed fact and the condition of the fuzzy rule “if A then B ” are exactly matched ($A' = A$), the consequence B will remain unchanged based on AARS. This is one of the advantages of AARS over CRI.

(2.) Reduction form:

In this form of ModF, the derived conclusion B' can be obtained by:

$$B' = SM \cdot B \quad \mu_{B'} = SM \cdot \mu_B \quad (7.10)$$

Figure 7.2(b) illustrates the effect of the reduction form of ModF. In this form, the membership values of the consequence B will be reduced based on the SM. In other words, the smaller the SM, the greater is the reduction of membership value of B . This imitates the Min operator’s contraction effect of CRI. Similar to the expansion form, in the case when the observed fact and condition in fuzzy rule are exactly identical ($A' = A$), a conclusion identical to consequence B will be derived. This can be considered as one of the advantages of AARS over CRI which usually produces a different result ($B' \neq B$).

Figure 7.2: Different forms of modification functions: (a) Expansion form; (b) Reduction form.

7.3 POPFNN-AARS

7.3.1 Structure

The singleton fuzzifier POPFNN-AARS proposed by Quek and Zhou (1999) is a five-layer network that employs neural network techniques to implement a fuzzy rule-based system using the singleton fuzzifier and the AARS fuzzy inference model together with the modified center average defuzzifier. Figure 7.3 shows the structure of singleton fuzzifier POPFNN-AARS. The five layers of this network are named as: Layer (1)- input layer, Layer (2)- condition layer, Layer (3)- rule-base layer, Layer (4)- consequence layer, and Layer (5)- output layer. For convenience POPFNN-AARS is denoted as POPFNN in this study.

The inputs and outputs of singleton fuzzifier POPFNN are represented as vectors $X^T = [x_1, x_2, \dots, x_i, \dots, x_{n1}]$ and $T = [y_1, y_2, \dots, y_m, \dots, y_{n5}]$, respectively; where $n1$ and $n5$ denote the number of the variables in the input and output layers, respectively. In the input layer, the singleton fuzzifier is used to fuzzify the crisp inputs (see Fig. 7.3). T_i in Fig. 7.3 denotes the number of labels for the i^{th} node in the input layer. Hence, the total number of nodes in the condition layer is $\sum_{i=1}^{n1} T_i$.

The symbol $n3$ in the rule-base layer denotes the number of fuzzy rules in the system. It is initially set to a maximum value of $\prod_{i=1}^{n1} T_i$. However, it is worth mentioning that this initial value could be potentially very big since it can be increased rapidly with an increase in the number of input and output variables. τ in Fig. 7.3 is the threshold applied to the fuzzy rules in the rule-base layer. Similarly to T_i , T'_m in the consequence layer represents the number of labels for the m^{th} node in the output layer. The detailed procedure of each layer is as follows:

Figure 7.3: Structure of singleton fuzzifier POPFNN.

Layer 1: The input layer

In this layer, the fuzzification process is accomplished. The net input f_i^I and net output o_i^I of the i th linguistic node L_i in the input layer of the singleton fuzzifier POP-FNN can be obtained by:

$$\text{net input: } f_i^I = x_i \quad \text{net output: } o_i^I = f_i^I \quad (7.11)$$

where x_i is the i^{th} element of the input vector X .

Layer 2: The condition layer

In this layer, the similarity measures between the fuzzified inputs and the input labels are calculated and then represented by the net outputs of the nodes. A sensitivity analysis done by Quek and Zhou (1999) showed that among five similarity measures introduced in Section 6.2.1, the Disconsistency Measure, Hausdorff Measure (*), and Hausdorff Measure (∞) are suitable for singleton fuzzifier POP-FNN. Since the similarity measures derived by Hausdorff Measure (*) and Hausdorff Measure (∞) are identical to each other, they will be referred to as Hausdorff Measures in the following paragraphs. The net input $f_{i,j}^H$ and net output $o_{i,j}^H$ of the input-label node $IL_{i,j}$ are given as:

$$\text{net input: } f_{i,j}^H = -\frac{(o_{i,j}^I - c_{i,j}^H)^2}{\delta_{i,j}^H} \quad \text{net output: } o_{i,j}^H = e^{f_{i,j}^H} \quad (7.12)$$

for the Disconsistency Measure and:

$$\text{net input: } f_{i,j}^H = |o_{i,j}^I - c_{i,j}^H| \quad \text{net output: } o_{i,j}^H = \frac{1}{(1 + f_{i,j}^H)} \quad (7.13)$$

for the Hausdorff Measures. $c_{i,j}^H$ and $\delta_{i,j}^H$ are the centroid and width of the membership function for the input-label node $IL_{i,j}$, respectively.

Layer 3: Rule-base layer

The neurons of this layer which are called rule nodes, represent the fuzzy rules. The output of this layer is dependent on the type of modification function chosen. The net input f_k^{III} and net output o_k^{III} for the rule node R_k of this layer are given for the expansion form:

$$\begin{aligned} \text{net input: } f_k^{III} &= \min_{i,j} (o_{i,j}^{II}) \\ \text{net output: } o_k^{III} &= \begin{cases} 1/f_k^{III} & f_k^{III} \geq \tau \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (7.14)$$

and reduction form:

$$\begin{aligned} \text{net input: } f_k^{III} &= \min_{i,j} (o_{i,j}^{II}) \\ \text{net output: } o_k^{III} &= \begin{cases} f_k^{III} & f_k^{III} \geq \tau \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (7.15)$$

where τ is the threshold applied to the fuzzy rules in this layer.

Layer 4: Consequence layer

Neurons of this layer are called output-label nodes and represent the labels of the corresponding output variables. In this layer, $\sum$ or $\cup$ operators can be used to calculate the height of the membership function of the final result at each output-label node. Therefore, the net input $f_{l,m}^{IV}$ and net output $o_{l,m}^{IV}$ of the output-label node $OL_{l,m}$ are given by:

$$\begin{aligned} \text{net input: } f_{l,m}^{IV} &= \left(\sum_k, \cup \right) o_k^{III} & \text{net output: } o_{l,m}^{IV} &= f_{l,m}^{IV} \end{aligned} \quad (7.16)$$

Quek and Zhou (1999) showed that the $\cup$ operator gives better results compared to $\sum$ operator; therefore, $\cup$ operator was used in the present study.

Layer 5: Output layer:

The neurons of this layer which are called defuzzification nodes represent the output variables. The net input f_m^V and net output o_m^V of the defuzzification node D_m in the output layer are given by:

$$\begin{aligned} \text{net input: } f_m^V &= \sum_{l=1}^{T'_m} (c_{l,m}^{IV} / \delta_{l,m}^{IV}) \times o_{l,m}^{IV} \\ \text{net output: } o_m^V &= \frac{f_m^V}{\sum_{l=1}^{T'_m} (o_{l,m}^{IV} / \delta_{l,m}^{IV})} \end{aligned} \quad (7.17)$$

where $c_{l,m}^{IV}$ and $\delta_{l,m}^{IV}$ are the centroid and width of the membership function for the output-label node $OL_{l,m}$, respectively.

7.3.2 Learning Algorithm

The learning process in POPFNN which adopts batch or offline learning consists of three phases: (1) self-organization for initialization of the parameters; (2) POP learning for rule identification, and (3) supervised learning for fine-tuning of the parameters.

Initialization Phase: Self-Organization

Membership functions of input and output-label nodes of POPFNN are important since the fuzzy information is stored in the membership functions. Correct determination of the center and width of these membership functions is required to capture and store the relevant information in the training data. The Kohonen's feature-map algorithm (Kohonen, 1989) is adopted in POPFNN to identify the

initial center and width of membership functions of both input and output variables. It should be noted that these centers and widths are initial values and are initial estimates of the clusters formed by input and output data samples. In the third phase of learning, the centers and widths will be fine-tuned by a supervised learning process.

Rule Identification Phase: POP Learning

To date, several approaches have been implemented to identify fuzzy rules. In some early fuzzy neural networks a self-organizing algorithm was used for this purpose (Hayashi et al., 1992; Yager, 1994). However, this approach seems to be inefficient for POP-FNN since the clusters in the joint input-output space are known to some extent after the initialization phase. In some other fuzzy neural networks, attempts were made to use the known membership function. As an example, competitive learning can be used to identify fuzzy rules (Lin and Lee, 1991). In this approach all possible fuzzy rules must be listed before the initiation of competitive learning. These fuzzy rules are then repeatedly tested against the training data to find their most likely consequences. The disadvantage of this approach is having iterative training to reach a stable state.

To more effectively utilize the information derived from the identified membership functions, POPFNN uses the Pseudo Outer-Product (POP) learning algorithm (Zhou and Quek, 1996) which is a simple one-pass algorithm. To understand the rule identification process in POP, consider a simple example of having two input variables x_1 and x_2 and output variable y_1 where $X^T = [x_1, x_2] \in D_1$, $Y^T = [y_1] \in D_2$. Suppose that after the self-organization phase, each linguistic variable has a label set of {small, medium, large} that is abbreviated as {S, M, L}. The POP learning algorithm considers all the possible rules (see Fig. 7.4) and each of these rules is fully connected to the output-label nodes in the consequence layer.

Figure 7.4: Sample POPFNN with two inputs and one output linguistic variable.

Once the membership functions have been determined, the set of training data $(X^{(r)}, Y^{(r)})$ is fed into both the input and output layers simultaneously. When $X^{(r)}$ is presented at the input layer, the membership values of each input-label node is derived using the membership functions that have earlier been determined during initialization phase. Subsequently, the firing strength f_k^{III} of the k th rule node in the rule-base layer will be produced using Eqns. (7.14) or (7.15). Similarly, $Y^{(r)}$ is fed into the output layer to calculate the membership values of the output-label nodes. For instance, if the training data satisfies the rule “if x_1 is large and x_2 is small then y_1 is medium” then the derived membership functions will have large values at the corresponding input-label nodes, namely: the input-label node “L” for linguistic

variable “ x_1 ”, the input-label node “S” for the linguistic variable “ x_2 ”, and the output-label node “M” for the defuzzification node “ y_1 ” (see Fig. 7.4). On the other hand, the membership values of the other input-label nodes in the condition layer as well as the output-label nodes in the consequence layer will be relatively small. Therefore, the firing strength of the rule which has the label “L” for the input variable “ x_1 ” and the label “S” for the input variable “ x_2 ” as its condition will be larger than other rules. If the weights of the links which connect the rule nodes and the output-label nodes are raised by the **product** of their firing strength and membership function values, then the improvement in the strength of the link between the rule node having conditions “ x_1 is L” and “ x_2 is S” and consequence “ y_1 is M” will be greater than other links. This is shown as a solid line in Fig. 7.4. Consequently, after training the fuzzy rules embedded in training data will have larger weights compared to other rules. The weights of the links can be obtained by:

$$w_{k,l,m} = \sum_{r=1}^p f_k^{\text{III}} \mu_{l,m}(y_m^{(r)}) \quad (7.18)$$

where $w_{k,l,m}$ is the weight of the link between the k th rule node and (l, m) th output-label node in the consequence layer; f_k^{III} is the firing strength of the k th rule node when presented with training data $X^{(r)}$; $\mu_{l,m}$ is the membership value derived at the (l, m) th output-label node when presented with training data $Y^{(r)}$; and $y_m^{(r)}$ is the m th element of the training data $Y^{(r)}$. It should be noted that the weights of all the links between the rule-base and the output layer are initially set to zero. After the POP learning process, the weights represent the strengths of the fuzzy rules having the corresponding output-label nodes as their real consequences. Amongst the links between a rule node and all the output-label nodes of a defuzzification node, the link with the largest weight is chosen and others deleted. In the situation when all the weights of the links between a rule node and the output-label nodes of a defuzzification node are very small or almost equal to each other, this shows that there is little or no relation to the output variable represented by that particular defuzzification node. Therefore, all the corresponding links can be deleted without affecting the outputs. From the removal of all links between a rule node and an output-label node it can be implied that the rule which is represented by that rule

node is irrelevant and hence should be eliminated. The whole learning process of POP is shown schematically in Figure 7.5.

Figure 7.5: Schematic process of identifying rules in POP learning algorithm.

Supervised Learning Phase: Back Propagation

In this phase, the well-known back-propagation algorithm (Werbos, 1988) is employed where the error signals are calculated and then are fed back to the system to adjust the centroid and width of membership functions for both output-label and input-label nodes.

7.3.3 Over-fitting

The threshold for firing a rule (τ in Section 7.3.1) affects rule pruning since in POPFNN the number of rules are not specified but are optimised during the learning process. Moreover, learning rate (LR) in back propagation algorithm is the other factor that can affect the learning process. Therefore, τ and LR are the two parameters that should be chosen properly to avoid over-fitting.

7.4 Methodology

7.4.1 Study sites and data used

Rainfall and runoff data collected from three different catchments ranging from a small experimental catchment (Catchment 1), a small urbanized catchment (Catchment 2), and a large rural watershed (Catchment 3) were analyzed for this study (see Section 3.5). As discussed in Section 6.3.3, the rainfall-runoff modeling in Catchments 1 and 2 is carried out on an event-based analysis due to the small size of the catchments. However, for Catchment 3 it is based on continuous time series. The results obtained from POP-FNN for Catchments 1, 2, and 3 are compared with the ones obtained by physically-based models KWM, SWMM, and URHM, respectively (see Section 3.3). Moreover, the results are also compared with ANFIS model (in which the CRI reasoning method is used) and also ARX model as a benchmark.

7.4.2 Input selection and model development

Following the results obtained in Section 4, Event 3 was chosen as the training event for POPFNN in Catchment 1. Using MICA (see Section 3.9.3) method for selecting the input candidates and conducting a verification procedure (with the same verification events used in Section 4), it was revealed that using two non-sequential rainfall inputs gives the best performance. The training/verification events and the selected input combination for Catchment 1 are summarized in Table 7.1. For Catchment 2, it was revealed that events with single peak in their discharge time series and a lag time close to the catchment lag time are the best choices to be chosen for training the model (see Section 5). Therefore, the same events used to train ANFIS4 (Section 5.3) were considered as the training events of POPFNN model in Catchment 2. Once again, MICA (Section 3.9.3) was used for selecting the input candidates and the best input combination was found through a verification process. Table 7.1 summarizes the training/verification events and the selected input combination for Catchment 2.

Table 7.1: Events and inputs used for POPFNN training in Catchments 1 and 2.

Catchment	Training Events	Verification Events	Inputs
1	27 Oct 2002	13 Nov 2002	R(t-4),R(t-8)
		18 Nov 2002	
		22 Nov 2002	
2	19 Jul 2005	17 Sep 2006	R(t-4),R(t-8)
	30 Jul 2005	04 Jul 2005	
	24 Dec 2005	16 May 2005	
		06 Apr 2006	

Rainfall-runoff modelling in Catchment 3 is based on a continuous time series. Splitting the data into training, verification, and testing parts was conducted similar to what adopted in Section 6. From approximately 16 years of available rainfall-runoff time series, 10 years (1 Jan 1996 to 21 Jan 2006) data was used for training, 1 year (1 Jan 1995 to 31 Dec 1995) for verification and 5 years (1 Jan 1990 to 31 Dec 1994) for testing. In the selection of training, verification, and the testing data, care was taken to ensure that extreme events were fairly distributed in all these three data sets. For Catchment 3, the three input selection methods of this study (Section 3.9) were used. The input selection analyses were firstly done considering rainfall antecedents only. Consistent to the findings for ANFIS and DENFIS models in Section 6, results obtained by POPFNN through verification stage revealed that using rainfall inputs only was not able to capture the rainfall-runoff process in this catchment (CE and r^2 values below 0.5 were obtained which are not satisfactory). This is attributed to the fact that there is only one rain gage in this large watershed (2,275 km²) and its location is very close to the outlet of the catchment (see Fig. 3.17) which makes the rainfall data not truly representative of the rainfall for the entire watershed. Therefore, it was decided to consider discharge antecedents in input selection procedure. Verification results showed that the input combination which included one discharge antecedent as input performed well. Details of the training, verification, and testing data as well as the inputs found for Catchment 3 during the verification phase and used for testing phase are provided in Table 7.2.

Table 7.2: Details of data and inputs used for POPFNN training in Catchment 3.

	Training Data	Verification Data	Testing Data	Inputs
Number of Days	3,674	365	1826	R(t-1), R(t-2), Q(t-1)
Starting Date	01 Jan 1996	01 Jan 1995	01 Jan 1990	
Closing Date	21 Jan 2006	31 Dec 1995	31 Dec 1994	

The parameters that are required as user inputs are the threshold (τ) (see Section 7.3), learning rate (LR) used in the supervised learning algorithm, the number of membership functions (labels) for each input and output, the types of the modification function (ModF), and the type of similarity measure (SM). τ is a threshold for firing a rule and hence affects rule pruning since in POPFNN the number of rules are not specified but are optimised during the learning process. τ and LR are the two parameters that should be chosen properly to avoid over fitting. Therefore, a Genetic Algorithm (GA) optimizer was coupled to the model POPFNN model and a sensitivity analysis was conducted. In this analysis, the training and verification data were used and the objective was maximizing the training performance in such a way that the verification performance is also reaches its the maximum performance. Table 7.3 shows the optimised parameters found for Catchments 2 and 3.

Table 7.3: Parameters of POPFNN model in Catchments 1, 2, and 3.

Catchment	ModF	SM	τ	LR	Number of Membership Functions for Inputs	Number of Membership Functions for Output
1	Reduction	Disconsistency Measure	0.11	0.410	2	2
2	Reduction	Disconsistency Measure	0.10	0.402	2	2
3	Reduction	Disconsistency Measure	0.01	0.336	3	3

7.5 Results and Discussion

POPFNN model results were compared with the results obtained by KWM for Catchment 1. A further simulation was also run with ANFIS for comparison between online and offline models. In the KW model, ϕ -index of 2.7 mm/hr corresponding to the loss rate for Event 6 was adopted in the simulation of the 10 testing events. This value of the ϕ -index was adopted as it was the median ϕ -index value for the 10 events (Chua et al., 2008). Figures 7.6(a) – (e) show the CE , r^2 , $RMSE$, MAE and RPE values, respectively, for the testing events obtained by the POPFNN model and compared with the results obtained by the ANFIS and KW models.

Figure 7.6: Comparison between different models used for Catchment 1. (a) CE , (b) r^2 , (c) $RMSE$, (d) MAE , and (e) RPE .

It can be seen that POPFNN results are comparable with the results obtained by the KWM and ANFIS models. As the experimental catchment is a completely impervious rectangular plane surface, the KWM is expected to model the events well since all the model parameters, except the loss rate, are known accurately. Thus, the good agreement between POPFNN and KWM results validates the performance of the POPFNN model for Catchment 1. Moreover, POPFNN estimates the peak discharge well for all events (all *RPE* values < 0.12). The predicted hydrographs for the 10 events (note that Event 3 is the training event) are shown in Fig. 7.7 and compared with the measured and KWM simulated hydrographs. It can be observed that the hydrographs simulated by the POPFNN model match the measured hydrographs well.

Figure 7.7: Hyetograph, observed and simulated hydrograph for events simulated by POPFNN and KWM in Catchment 1.

For further comparison the average values of CE , r^2 , $RMSE$, MAE and RPE values over all events obtained by POPFNN, ANFIS, KWM, and an ARX model (the same ARX model presented in Section 4.7) are compared in Table 7.4. In general, POPFNN outperforms the ARX model and is comparable to ANFIS and KWM models. POPFNN model results were compared with the results obtained by SWMM (SWMM1 in Section 5) based on the study by Tan et al., (2008) for Catchment 2. Figures 7.8(a) – (e) show the box plots of CE , r^2 , $RMSE$, MAE and RPE for the 63 events simulated by POPFNN and SWMM. Based on this analysis, it is shown that POPFNN model results are comparable to SWMM and almost identical to results obtained by ANFIS. POPFNN model outperforms other models in peak estimation since the median $RPE = 0.160$ which is less than the error obtained by ANFIS (median $RPE = 0.238$) and SWMM (median $RPE = 0.357$) models.

Table 7.4: Comparison of POPFNN, ANFIS, KWM, and ARX models for Catchment 1.

Model	CE	r^2	$RMSE$ (l/s)	MAE (l/s)	RPE
POPFNN	0.966	0.967	0.028	0.018	0.067
ANFIS	0.950	0.957	0.034	0.022	0.060
KWM	0.952	0.965	0.030	0.018	0.089
ARX(0,6,3)	0.944	0.952	0.040	0.023	0.090

Qualitative assessment of the predicted hydrographs was carried out by comparing the predicted and measured hydrographs of a representative event for each of the models in Figs. 7.9(a) and 7.9(b). The 04 Jul 2005 event was chosen as representative of the POPFNN model while the 16 Apr 2005 event was chosen as representative of the SWMM model. These are events which produced the median $RMSE$ for the individual models. Figures 7.9(a) and 7.9(b) show that in general, the POPFNN model is able to reproduce the hydrograph shape reasonably well. However, there are indications of fluctuations in the rising and recessing limb of the hydrographs produced by the POPFNN model.

The average values of CE , r^2 , $RMSE$, MAE and RPE for the 63 events of Catchment 2 simulated by POPFNN, ANFIS, SWMM, and ARX models are compared in Table 7.5. It should be noted that ANFIS, SWMM, and ARX models in Table 7.5 are the same as those discussed in Section 5.3.3). As can be seen, results obtained by POPFNN are comparable with SWMM models. POPFNN slightly outperforms ANFIS while it is significantly superior to ARX model. POPFNN is the best model in peak estimation since the average RPE value for this model is significantly smaller than other models.

Figure 7.8: Comparison between different models used for Catchment 2. (a) CE , (b) r^2 , (c) $RMSE$, (d) MAE , and (e) RPE .

Figure 7.9: Hyetograph, observed and simulated hydrograph for median events simulated by POPFNN and SWMM for Catchment 2.

Table 7.5: Comparison of POPFNN, ANFIS, SWMM, and ARX models for Catchment 2.

Model	CE	r^2	$RMSE$ (m^3/s)	MAE (m^3/s)	RPE
POPFNN	0.632	0.709	2.519	1.754	0.169
ANFIS	0.570	0.720	2.970	1.933	0.229
SWMM	0.642	0.770	2.648	1.719	0.348
ARX(0,6,4)	0.423	0.595	3.248	2.158	0.358

The results of the POPFNN model are compared with results obtained from the URHM model, ANFIS, and ARX model for Catchment 3 in Table 7.6. Similar to Section 6, ANFIS model uses $R(t)$, $R(t-1)$, $R(t-2)$, and $Q(t-3)$ as inputs and two Gaussian membership functions are allocated to each input. The ARX model is calibrated following the approach discussed in Section 4.7. As can be seen, POPFNN gives slightly better performance when compared with URHM model since the CE and r^2 values are higher and $RMSE$ and MAE are lower, compared to URHM model results. In contrast to the results for Catchments 1 and 2 where POPFNN and ANFIS results were almost identical, POPFNN performed slightly

worse than ANFIS for Catchment 3, while outperforming the ARX model. The distribution of RPE determined for 27 peak discharges where $Q_{\text{observed}} > 100 \text{ m}^3/\text{s}$ is shown in Fig. 7.10. POPFNN performs best in peak discharge estimation (average = 0.220, median = 0.231) when compared with URHM (average = 0.296, median = 0.277) and ANFIS (average = 0.255, median = 0.272) models. Moreover the RPE for POPFNN spreads over a slightly smaller range of [0.007, 0.514] compared to [0.022, 0.691] for URHM and [0.005, 0.577] for ANFIS.

Table 7.6: Comparison of CE , r^2 , MAE , and $RMSE$ values for POPFNN, ANFIS, URHM, and ARX models for Catchment 3.

Model	CE	r^2	$RMSE$ (m^3/s)	MAE (m^3/s)
POPFNN	0.791	0.792	19.536	6.624
ANFIS	0.840	0.850	17.133	6.550
URHM	0.758	0.760	21.052	11.189
ARX(3,3,0)	0.736	0.736	21.962	8.992

Figure 7.10: Comparison of RPE values for peak discharges $> 100 \text{ m}^3/\text{s}$ for Catchment 3.

Figure 7.11 shows the discharge time series simulated by POPFNN for Catchment 3 which is compared with the discharge time series predicted by the URHM model and the observed value. The comparison shows that the POPFNN model is able to predict both the high and low flows well and the predicted hydrograph is comparable to the URHM model result. A notable exception is the under prediction of the highest discharge recorded in the testing dataset (time = 1,629 days). This highlights the problem of extrapolation with data driven methods, in general, since this class of models do not work as well if the peak discharge to be simulated far exceeds the peak discharge used in training the model. It was possible to include the highest discharge value in the dataset used to train the POPFNN model (which would have resulted in an improvement in results), but this was avoided since the intention was to use the same dataset for training both the POPFNN and URHM models, in order to facilitate a more realistic comparison between the models. Nevertheless, the *RPE* obtained by POPFNN for this event was 0.162, which is comparable with the URHM model where an *RPE* of 0.178 obtained. This highlights the general difficulty in making accurate predictions of extreme events faced by all models. Lastly, it is worth noting that the POPFNN model used discharge antecedents in its inputs. As mentioned in Section 6.4, this was inevitable since the rainfall recorded at Beimen (Fig. 3.16) was not truly representative of the rainfall for the entire watershed as it is not centrally located. As highlighted in Section 6.4, the URHM model utilised additional (August 1, 2004 to January 21, 2006) rainfall data obtained from independent sites in its calibration (Qin et al., 2010) which was not made available for our study. This highlights an important distinction between data driven models (such as ANFIS, DENFIS, and POPFNN) and URHM models. The URHM model requires accurate rainfall inputs since precipitation is the primary driver for the model; POPFNN (identical to DENFIS), on the other hand does not need complete precipitation information but the trade-off is that measured discharge is required.

Figure 7.11: Measured rainfall, observed discharge, and simulated discharges by POPFNN and URHM models for Catchment 3.

As seen in the results for the three contrasting catchments, POPFNN and ANFIS showed almost similar performance in terms of the error statistics considered. However, the rule creation procedure for POPFNN and ANFIS are different. For ANFIS, which is a Takagi-Sugeno-type NFS, the output is a crisp value which is defined by a linear function while in POPFNN, which is a Mamdani-Type NFS, the output is defined by fuzzy sets. Therefore, if n membership functions are specified for each of the input variables, n^q rules where q is the number of input variables will be produced in ANFIS while $n^q \times m$ rules where m is the number of membership functions for each output variable will be produced in POPFNN. In most batch learning models, n has to be selected by trial and error. Although the number of rules can be increased by the specification of greater number of membership functions, this can result in over-fitting leading to deterioration in the model results. On the other hand, decreasing the number of rules by choosing a smaller number of membership functions can also worsen model performance due to an insufficient number of rules required to properly capture the associations between inputs and output variables. POPFNN learning algorithm incorporates a rule pruning mechanism that begins with a large number $n^q \times m$ of rules, with the unused or less important rules removed during the training process (Section 7.3.2). This represents an advantage over the learning mechanism used in ANFIS where the number of rules is determined by the user from the specification of the number of member functions, where an improper choice may lead to redundant rules that may result in a deterioration in model performance. In contrast, POPFNN has the ability to select only the essential rules, via the rule pruning procedure, from all possible rules. Thus, the deteriorating effect of having redundant or an excessive number rules is avoided.

The rules for Catchment 1, obtained by the POPFNN and ANFIS models are shown in Tables 7.7 and 7.8, respectively. For ANFIS, 2 membership functions were used for each of the two inputs and a linear function was chosen for the output, therefore, $2^2 = 4$ rules were produced. In POPFNN, however, 2 membership functions were chosen for each of the inputs and output variables; therefore, $2^2 \times 2 = 8$ rules were initially produced and then reduced to 4 rules by the pruning mechanism in POPFNN.

Table 7.7: Extracted rules for Catchment 1 from POPFNN model (L and H represent low and high values, respectively).

Rule Number	Input X_1	Input X_2	Output Y
1	L	L	L
2	L	H	L
3	H	L	H
4	H	H	H

Table 7.8: Extracted rules for Catchment 1 from ANFIS model with 2 membership functions for inputs and linear function for output (L and H represent low and high values, respectively).

Rule Number	Input X_1	Input X_2	Output: $Y=aX_1+bX_2+c$		
			a	b	c
1	L	L	1.213	0.548	-0.069
2	L	H	-2.797	0.172	0.043
3	H	L	1.467	-1.140	-0.026
4	H	H	-5.228	-0.851	5.153

The rules for Catchment 2, obtained by the POPFNN and ANFIS models are shown in Tables 7.9 and 7.10, respectively. Similar to Catchment 1, 2 membership functions were used for each of the two inputs and a linear function was chosen for the output, therefore, i.e. $2^2 = 4$ rules were produced for Catchment 2. In POPFNN, however, 2 membership functions were chosen for each of the inputs and output variables; therefore, $2^2 \times 2 = 8$ rules were initially produced which was then reduced to 4 rules by the pruning algorithm in POP-FNN.

Table 7.9: Extracted rules for Catchment 2 from POPFNN model (L and H represent low and high values, respectively).

Rule Number	Input X_1	Input X_2	Output Y
1	L	H	H
2	H	H	H
3	H	L	L
4	L	L	L

Table 7.10: Extracted rules for Catchment 2 from ANFIS model with 2 membership functions for inputs and linear function for output (L and H represent low and high values, respectively).

Rule Number	Input X_1	Input X_2	Output: $Y=aX_1+bX_2+c$		
			a	b	c
1	L	L	-2.548	9.526	-0.588
2	L	H	-6.480	9.493	-5.055
3	H	L	-2.874	36.462	-1.545
4	H	H	-4.420	35.001	-18.605

Tables 7.11 and 7.12 show the rules produced by POPFNN and ANFIS models for Catchment 3, respectively. For ANFIS, 2 membership functions were used for each input variable and the output function was the linear function; therefore, $2^4 = 16$ rules were produced. However, in POPFNN, 3 membership functions were used for each of the inputs and output variables which initially produced $3^3 \times 3 = 81$ rules. The pruning algorithm then reduced the number of rules to 25. As shown in Table 7.11, some rule outputs contain two labels (for example L, M in Rule 1). This indicates that the output in these rules is a combination between the values resultant from two membership functions. This ability of the POPFNN model to produce an output which is a combination of two membership functions adds flexibility to the POPFNN model. It should be noted that all the parameters of

the output functions for ANFIS (Tables 7.8, 7.10, 7.12) are based on standardized values. The consequent part of the rule by fuzzy sets in POPFNN (Tables 7.7, 7.9, and 7.11), however, is a linguistic value. This is useful in the case where a qualitative understanding of the outputs is necessary. For example, the meaning of Rule 8 produced by POPFNN is: “*If the rainfall in present day is low, rainfall in previous day is high, and discharge in previous day is medium then discharge in present day is medium*”. However, in a rule obtained by ANFIS, the output is given as a linear function of inputs; however, it is not easy to extract any linguistic meaning from such linear functions. This highlights the advantage of linguistic models such as POPFNN over precise models such as ANFIS.

Table 7.11: Extracted rules for Catchment 3 from POPFNN model (L, M, and H represent low, medium, and high values, respectively).

Rule Number	Input X ₁	Input X ₂	Input X ₃	Output Y
1	L	L	L	L,M
2	L	L	M	L,M
3	L	L	H	M
4	L	M	L	L,M
5	L	M	M	L,M
6	L	M	H	M,H
7	L	H	L	L,M
8	L	H	M	M
9	L	H	H	M,H
10	M	L	L	L,M
11	M	L	M	L,M
12	M	M	L	L,M
13	M	M	M	L,M
14	M	M	H	M,H
15	M	H	L	L
16	M	H	M	M
17	M	H	H	H
18	H	L	L	L,M
19	H	L	M	M,H
20	H	L	H	H
21	H	M	L	M
22	H	M	M	H
23	H	H	L	M,H
24	H	H	M	M,H
25	H	H	H	H

Table 7.12: Extracted rules for Catchment 3 from ANFIS model (L and H represent low and high values, respectively).

Rule No.	Input X_1	Input X_2	Input X_3	Input X_4	Output: $Y=aX_1+bX_2+cX_3+dX_4+e$				
					a	b	c	d	e
1	L	L	L	L	0.063	0.115	11.700	0.798	-1.132
2	L	L	L	H	-3.095	4.007	0.321	0.757	-0.416
3	L	L	H	L	0.751	-0.198	12.280	1.459	-10.788
4	L	L	H	H	-3.956	17.809	0.785	0.624	-2.228
5	L	H	L	L	-2.306	0.679	0.135	1.311	0.017
6	L	H	L	H	-6.892	12.103	4.774	-3.903	-4.736
7	L	H	H	L	6.351	-0.014	-0.427	15.919	-2.170
8	L	H	H	H	9.592	0.043	-24.076	34.505	-21.257
9	H	L	L	L	0.504	1.441	1.781	0.533	-0.379
10	H	L	L	H	-4.491	15.158	-7.253	-3.063	4.152
11	H	L	H	L	3.094	-7.029	-0.790	-7.272	0.186
12	H	L	H	H	-14.844	11.740	13.198	8.369	6.288
13	H	H	L	L	-1.169	0.175	-9.240	30.976	-1.209
14	H	H	L	H	15.513	-66.984	-19.440	19.208	1.328
15	H	H	H	L	-11.841	5.052	9.994	42.102	1.423
16	H	H	H	H	12.745	-5.931	26.115	-10.004	7.854

7.6 Conclusions

The following can be concluded from this study:

- (i) Pseudo Outer Product Fuzzy Neural Network (POPFNN) was tested against physically-based models for 3 catchments, with different sizes and different modelling complexities. Generally, POPFNN was comparable if not superior, in some respects, compared to the KW, SWMM, and URHM used for Catchments 1, 2, and 3, respectively.
- (ii) Although the improvement in results obtained by the rule pruning mechanism used in the POPFNN learning algorithm over ANFIS which has a fixed number of rules was modest, POPFNN has a more flexible rule structure with

an optimal number of rules to better capture the association between input and output.

- (iii) Results obtained by POPFNN for Catchments 1, 2, and 3 were compared with the ones obtained by ARX model as a bench mark. It was revealed that POPFNN is superior to ARX models in all catchments.
- (iii) The Mamdani-type FIS used in POPFNN is able to provide a linguistic understanding for output variable in the consequent part of the rules while the Takagi-Sugeno-type FIS used in ANFIS specifies the rule consequents as quantitative or crisp values. This linguistic nature of the Mamdani-type FIS could be advantageous if a qualitative understanding for the output variable is necessary.

CHAPTER 8 DISCUSSION

The detailed discussions on specific topics were provided earlier in related chapters. However, this chapter aims to provide a broad discussion of the general perspectives and conclusions reached in this study and also put them in an overall context. Moreover, limitations of this study are pointed out and topics of future studies to extend this study are also discussed. Each section of this chapter refers to one of the main chapters of this study.

8.1 Input Selection in Event-Based R-R Modeling

Three different input selection methods namely Correlation Analysis (CA), Mutual Information-Correlation Analysis (MICA), and Stepwise Regression (SR) were adopted in the selection of the input candidates for ANFIS model in Catchment 1 (Chapter 4). Comparing the three types of common rainfall input combinations known as sequential, pruned sequential and non-sequential rainfall time series revealed that non-sequential rainfall inputs generally give better performance compared to the other choices. Moreover, it was found that MICA gave the best results in choosing the best input candidates. This finding was examined for Catchment 2 in Chapter 5 and the results obtained were consistent with what was found in Catchment 1. Therefore, this indicates that non-sequential rainfall inputs are the most appropriate input choices in event-based R-R modeling. However, it is noted that these findings are based on two small catchments from a tropical area. Therefore, further study on examining these findings in other catchments would be needed to support the conclusions obtained.

The use of non-sequential inputs can be supported if one considers the fact that simulation of different parts of hydrograph such as rising and recessing limbs requires different types of information based on the lag effect between time of rainfall and the occurrence of runoff. The use of non-sequential rainfall inputs thus optimizes the input information required by rainfall antecedents at the appropriate lags that contribute to various portions of the hydrograph. Thus, the rainfall input that is closer to the start of the rainfall event may contain information pertinent to the rising phase of the hydrograph, compared a second rainfall input chosen at a

later part of the rainfall hyetograph, which could be more important for associations with the recession phase of the hydrograph.

8.2 Training Event Selection in Event-Based R-R Modeling

Hydrograph shape was considered as a measure in training event selection for ANFIS in Chapter 5 (Section 5.2). The results revealed that events with a single peak in their discharge time series are better choices to be chosen for training the ANFIS model in Catchment 2 compared to the other events. This result was also assessed for Catchment 1 and the results were consistent with the ones obtained earlier. Therefore, for event-based analysis, events with one peak (or at most two peaks as shown in some cases for Catchment 1) in their hydrograph are appropriate choices for training an NFS model. This could be attributed to the fact that the hydrograph with multiple peaks could be a result of multiple events that are integrated. It could be expected that capturing the association between each individual rainfall event and the resulting discharge is not easily learned by the model.

The effect of lag time on training event selection was discussed in Section 5.3. This study showed that events belong to a group of events with a specific lag time can be simulated best when the model is trained with the events from the same group. However, this finding may not be practical since the lag time of the event is not known *a priori*. To extend this finding in training event selection for practical purposes, it was shown that training the model with events of lag time close to the median lag of the available events can be useful, since the lag of the majority of the events fall around that value.

8.3 Adaptable Neuro-Fuzzy Systems

A neuro-fuzzy model such as ANFIS which utilizes offline learning cannot be adapted (or modified) to possible changes that may occur. This is attributed to the fact that the learning is limited to the information captured from a training data set based on historical events. For a NFS to be adaptable, learning should take place during the testing phase as well. This necessitates the use of online models, in

contrast to batch learning models, on which ANFIS is based. Therefore, a Takagi-Sugeno type neuro-fuzzy system with online learning, DENFIS, was investigated in Chapter 6.

In the first stage of this study, the capabilities of DENFIS in R-R modelling for three different catchments were compared with ANFIS and some other benchmarks. The results from DENFIS were consistent with ANFIS and physically-based models' results and superior to regression-based model ARX which was chosen as benchmark. It should be mentioned that, both DENFIS and ANFIS models in Catchment 3 benefited from having discharge antecedents in their inputs. Therefore, the fairness of the comparison of these two models with URHM model could be questioned. However, the URHM model utilised additional (August 1, 2004 to January 21, 2006) rainfall data obtained from independent sites in its calibration (Qin et al., 2010) which was not made available for our study. In general, results showed that online learning can perform well for R-R modelling; however, no significant improvement was found over offline learning method employed in ANFIS. However, it was found that DENFIS could be trained with less data compared to ANFIS, and this could be considered as an advantageous feature, when data is limited. In addition, DENFIS was found to be a better model compared to ANFIS since there is no need to find the suitable number of membership functions by trial. This could be advantageous when the number of inputs is large and many combinations of the number of membership functions to be allocated to each input should be tried.

In the second stage of this study, the online learning and evolving algorithm of the DENFIS were used to develop a real-time version of DENFIS (RT-DENFIS) where the NFS is able to learn during testing. The proposed RT-DENFIS model could be considered as an adaptive model to any possible changes in the R-R process since this model continues learning automatically in a point by point manner. The capabilities of the proposed model were evaluated by the large amount of historical data (43 years) of a large rural catchment (Catchment 4). RT-DENFIS was able to model the R-R process over a longer period of time, without the need to be retrained, while ANFIS needed periodic retraining to achieve similar results.

A noteworthy difference between DENFIS (or RT-DENFIS) and ANFIS models is that the DENFIS model requires a larger number of rules. This is because ANFIS employs grid partitioning algorithm which utilizes the whole training data set in advance while the rule base in DENFIS evolves based on presentation of each data point to the model. However, the current learning algorithm in DENFIS is based on incremental learning and no pruning mechanism is considered. Therefore, this could be the subject of a future study, to add a pruning mechanism to DENFIS algorithm in order to remove unnecessary rules.

8.4 Interpretable Neuro-Fuzzy Systems

Takagi-Sugeno type NFS are precise models. Such models lack interpretability. This is because the consequent part of the rules in precise models are linear functions and it is difficult to extract linguistic meanings from these rules and connect them to the physical inputs. Therefore, a linguistic model known as POPFNN was chosen to be tested for the same catchments used in the analysis with the precise models. The analysis showed that POPFNN is comparable with ANFIS and the physically-based models used and superior to regression-based models. The improvements of POPFNN in Catchments 1 and 2 were modest while in Catchment 3 ANFIS performed slightly better than POPFNN. However, the purpose of this study was not to achieve a better model in terms of the error estimates but rather the target was to achieve better interpretability compared to ANFIS, while maintaining acceptable good model performance. This study thus suggests that when there is a need for linguistic understanding, for example in water resources management problems (ground water table management, reservoir water level management, etc.), models like POPFNN can be useful since not only does POPFNN provide acceptable results, but also these results are based on rules that could be better interpreted.

One of the main issues of POPFNN model is the complexity of the structure which can make it difficult to use. Several parameters should be tuned and the sensitivity of the model to these parameters is high. For developing the model for any problem (for example each catchment of this study) a lot of effort was needed

to set the model's parameters. Due to time limitations, a sensitivity analysis to evaluate the effect of the parameters on model performance could not be conducted. This could be the subject of future studies. The other issue with the POPFNN model is in selecting the proper number of labels for each input and output variable. A small number of labels can affect the learning by having an insufficient number of rules, while a large number of labels can cause the model structure and rule-base to become unnecessarily complex, which in turn will lead to increased runtime. In general, a trial and error procedure is used to choose the appropriate number of labels. However, in case of having several inputs and outputs this procedure could become tedious and time consuming.

Although POPFNN is a more interpretable model compared to ANFIS or DENFIS, it is still an offline model. One of the suggested extensions to this study could be adopting an online learning in the structure of POPFNN. In such case, the desired model not only is more interpretable, but also will be adaptive. It is worth mentioning that POPFNN benefits from a rule pruning algorithm which can make it a superior model compared to DENFIS with incremental learning.

CHAPTER 9 CONCLUSIONS

Neuro-Fuzzy models represent the current state of the art applications of computer intelligence tools in rainfall-runoff modeling. Yet, in spite of the growing interest in the use of neuro-fuzzy systems in hydrological modeling, a proper understanding of the behavior of these models is sorely lacking in the literature. Indeed, most papers written on the subject focus on the application aspects, which are important in establishing the suitability of NFS as modeling tools; however, unless some of the basic questions with regards to the use of these models are addressed, it will be difficult for NFS to gain widespread acceptance as alternative modeling tools in hydrology. This thesis has focused on four important issues, listed in Chapter 1 as the objectives of this research. The following are conclusions obtained from this research:

- (1) *Input selection for neuro-fuzzy systems such as ANFIS in event-based rainfall-runoff modeling*
 - (1.a) The results from a study on input selection showed that models using only rainfall antecedents as inputs are superior in long-term discharge forecasts while models which included $Q(t-1)$ as input were better in short-term discharge predictions. However, discharge antecedents were found to be the source of time shift error in discharge forecasts. In addition, non-sequential rainfall antecedents were found to be a better choice as inputs compared to sequential and pruned sequential rainfall antecedents, which are commonly found in the literature.
 - (1.b) Among several common input selection methods tested, mutual information and correlation analyses (MICA) was found to be suitable for event-based rainfall-runoff modeling when tested using data derived from an artificial catchment with an ANFIS model. This was later confirmed when MICA was applied to a real catchments with two other NFS models, DENFIS and POPFNN. From the studies carried out, it was found that MICA is the most reliable method and the use of non-sequential rainfall inputs is appropriate for event-based R-R modeling.
 - (1.c) Moreover, sequential and pruned sequential rainfall inputs are less likely to give reasonably good results compared to non-sequential rainfall inputs in event-based R-R modeling.

- (2) *Training event selection for neuro-fuzzy systems in event-based rainfall-runoff modeling*
- (2.a) The results of this study showed that training events with one peak in the hydrograph are most suited for NFS model training. The worst results were obtained when the model was trained by events with multiple peaks in the hydrograph. This finding was confirmed for both Catchments 1 and Catchment 2.
- (2.b) Lag time was found to be an important factor in training event selection. Events simulated by models trained with training events with the same lag time gave the best results. The worst results were obtained when the lag time of the testing events were significantly different from the lag time of the training events. However, in practice the lag time of an event is not known *a priori*. Therefore, as a practical application of this finding, it was suggested that the NFS model should be developed based on historical events which had the median lag time.
- (3) *Adaptable neuro-fuzzy systems (NFS) for R-R modeling*
- (3.a) A Takagi-Sugeno neuro-fuzzy system DENFIS, which employs an online learning algorithm, was found to perform well for event-based (as well as continuous) rainfall-runoff modeling over a range in catchment scales. The results obtained by DENFIS were comparable results with the ones obtained by physically-based models and ARX as the benchmark. This highlighted that the online learning is suitable for use with NFS in R-R modeling.
- (3.b) Although the improvement in results obtained by the local learning algorithm over the global offline learning algorithm used in ANFIS were modest, DENFIS was considered as a better model compared to ANFIS since DENFIS is able to optimise the number of membership functions required. In addition, DENFIS requires lesser data to train, compared to ANFIS.
- (3.c) Real time DENFIS (RT-DENFIS) was developed and compared with ANFIS. It was revealed that RT-DENFIS gave consistently better results without the need for retraining (since the model is constantly updated); however, batch models trained with historical data such as ANFIS had to be retrained periodically in order to achieve reasonably good predictions. RT-DENFIS was also compared with ARX and Naïve models as bench marks. RT-DENFIS is applicable in forecasting problem in which R-R process within the catchment is expected to change, for example due to the land use change or changes in climate.

- (4) *Towards interpretable NFS for R-R modeling*
- (4.a) A more advanced NFS based on similarity measures known as POPFNN-AARS (simply denoted as POPFNN) was introduced for event-based (and continuous) rainfall-runoff modeling for three different sized catchments. The results obtained by POPFNN were comparable with the ones obtained from physically-based models and ARX model as bench mark.
- (4.b) POPFNN was found to have a more flexible rule structure with an optimal number of rules. This is attributed to the pruning mechanism of POPFNN which is able to find the proper number of rules during training.
- (4.c) The Mamdani-type FIS used in POPFNN was found to be able to provide a linguistic representation for the output variable in the consequent part of the rules. This is in contrast to the Takagi-Sugeno-type FIS used in ANFIS which specifies the rule consequents as quantitative or crisp values. The linguistic nature of the Mamdani-type FIS is advantageous if a qualitative understanding for the input-output association is necessary.

In conclusion, this study has shed light on some issues on event-based R-R modeling by NFS models. Moreover, seeking to improve the two common weak points of data driven models known as adaptability and interpretability, alternative models are suggested and evaluated for R-R modeling in different catchment sizes.

REFERENCES

- 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.
- Abrahart, R.J., Kneale, P.E., See, L.M., 2004. *Neural networks for hydrological modelling*. Taylor & Francis.
- Achela, D., Fernando, K., Jayawardena, A.W., 1998. Runoff forecasting using RBF networks with OLS algorithm. *Journal of Hydrologic Engineering*, 3(3): 203-209.
- Agarwal, A., Singh, R.D., 2004. Runoff modelling through back propagation artificial neural network with variable rainfall-runoff data. *Water Resources Management*, 18(3): 285-300.
- Akaike, H., 1974. New look at the statistical model identification. *IEEE Transactions on Automatic Control*, AC-19(6): 716-723.
- Alley, W.M., Dawdy, D.R., Schaake Jr, J.C., 1980. Parametric-deterministic urban watershed model. *Journal of the Hydraulics Division, ASCE*, 106(HY5, Proc. Paper, 15392): 679-690.
- Anctil, F., Perrin, C., Andréassian, V., 2004. Impact of the length of observed records on the performance of ANN and of conceptual parsimonious rainfall-runoff forecasting models. *Environmental Modelling and Software*, 19(4): 357-368.
- Ang, K.K., Quek, C., 2005. RSPOP: Rough Set-Based Pseudo Outer-Product Fuzzy Rule Identification Algorithm. *Neural Computation*, 17(1): 205-243.
- Antar, M.A., Ellassiouti, I., Allam, M.N., 2006. Rainfall-runoff modelling using artificial neural networks technique: A Blue Nile catchment case study. *Hydrological Processes*, 20(5): 1201-1216.
- Aqil, M., Kita, I., Yano, A., Nishiyama, S., 2007. A comparative study of artificial neural networks and neuro-fuzzy in continuous modeling of the daily and hourly behaviour of runoff. *Journal of Hydrology*, 337(1-2): 22-34.
- Bowden, G.J., Dandy, G.C., Maier, H.R., 2005. Input determination for neural network models in water resources applications. Part 1 - Background and methodology. *Journal of Hydrology*, 301(1-4): 75-92.
- Castellano-Méndez, M., González-Manteiga, W., Febrero-Bande, M., Manuel Prada-Sánchez, J., Lozano-Calderón, R., 2004. Modelling of the monthly and daily behaviour of the runoff of the Xallas river using Box-Jenkins and neural networks methods. *Journal of Hydrology*, 296(1-4): 38-58.

- Chang, P.C., Liu, C.H., 2008. A TSK type fuzzy rule based system for stock price prediction. *Expert Systems with Applications*, 34(1): 135-144.
- Chen, S.H., Lin, Y.H., Chang, L.C., Chang, F.J., 2006. The strategy of building a flood forecast model by neuro-fuzzy network. *Hydrological Processes*, 20(7): 1525-1540.
- Cheng, X., Noguchi, M., 1996. Rainfall-runoff modelling by neural network approach, *Proc. of Int. Conf. on Water Resour. & Environ. Res.*, Kyoto, Japan, pp. 143-150.
- Chiang, Y.M., Chang, L.C., Chang, F.J., 2004. Comparison of static-feedforward and dynamic-feedback neural networks for rainfall-runoff modeling. *Journal of Hydrology*, 290(3-4): 297-311.
- Cho, S.Y., Quek, C., Seah, S.X., Chong, C.H., 2009. HebbR2-Taffic: A novel application of neuro-fuzzy network for visual based traffic monitoring system. *Expert Systems with Applications*, 36(3 PART 2): 6343-6356.
- Chua, L.H.C., Wong, T.S.W., Sriramula, L.K., 2008. Comparison between kinematic wave and artificial neural network models in event-based runoff simulation for an overland plane. *Journal of Hydrology*, 357(3-4): 337-348.
- da CostaCouto, M.P., 2009. Review of input determination techniques for neural network models based on mutual information and genetic algorithms. *Neural Computing and Applications*, 18(8): 891-901.
- Dawson, C.W., Harpham, C., Wilby, R.L., Chen, Y., 2002. Evaluation of artificial neural network techniques for flow forecasting in the River Yangtze, China. *Hydrology and Earth System Sciences*, 6(4): 619-626.
- Dawson, C.W., See, L.M., Abraham, R.J., Heppenstall, A.J., 2006. Symbiotic adaptive neuro-evolution applied to rainfall-runoff modelling in northern England. *Neural Networks*, 19(2): 236-247.
- Dawson, C.W., Wilby, R., 1998. An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal*, 43(1): 47-66.
- Dawson, C.W., Wilby, R.L., 1999. A comparison of artificial neural networks used for river flow forecasting. *Hydrology and Earth System Sciences*, 3(4): 529-540.
- Dawson, C.W., Wilby, R.L., 2001. Hydrological modelling using artificial neural networks. *Progress in Physical Geography*, 25(1): 80-108.
- de Vos, N.J., Rientjes, T.H.M., 2007. Correction of timing errors of artificial neural network rainfall-runoff models. *Hydroinformatics in Practice Computational Intelligence and Technological Developments in Water Applications*.

- Firat, M., Güngör, M., 2008. Hydrological time-series modelling using an adaptive neuro-fuzzy inference system. *Hydrological Processes*, 22(13): 2122-2132.
- Fraser, A.M., Swinney, H.L., 1986. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33(2): 1134-1140.
- Furundzic, D., 1998. Application example of neural networks for time series analysis: Rainfall-runoff modeling. *Signal Processing*, 64(3): 383-396.
- Gaume, E., Gosset, R., 2003. Over-parameterisation, a major obstacle to the use of artificial neural networks in hydrology? *Hydrology and Earth System Sciences*, 7(5): 693-706.
- Goodwin, G.C., Sin, K.S., 1984. *Adaptive Filtering Prediction and Control*. Prentice-Hall.
- Govindaraju, R.S., 2000. Artificial neural networks in hydrology. II: Hydrologic applications. *Journal of Hydrologic Engineering*, 5(2): 124-137.
- Halff, A.H., Halff, H.M., Azmoodeh, M., 1993. Predicting runoff from rainfall using neural networks, *Proc., Engrg. Hydrol., ASCE*, New York, pp. 760-765.
- Haugh, L.D., Box, G.E.P., 1977. Identification of dynamic regression (distributed lag) models connecting two time series. *Journal of the American Statistical Association*, 72(397): 121-130.
- Hayashi, I., Nomura, H., Yamasaki, H., Wakami, N., 1992. Construction of fuzzy inference rules by NDF and NDFL. *International Journal of Approximate Reasoning*, 6(2): 241-266.
- Haykin, S., 1999. *Neural networks: a comprehensive foundation*. Prentice Hall.
- He, J., Valeo, C., Chu, A., Neumann, N.F., 2011. Prediction of event-based stormwater runoff quantity and quality by ANNs developed using PMI-based input selection. *Journal of Hydrology*, 400(1-2): 10-23.
- Hjelmfelt, A.T., Wang, M., 1993. Artificial neural network as unit hydrograph applications, *Proc., Engrg. Hydrol., ASCE*, New York, pp. 754-759.
- Hong, Y.S.T., 2012. Dynamic nonlinear state-space model with a neural network via improved sequential learning algorithm for an online real-time hydrological modeling. *Journal of Hydrology*, 468-469: 11-21.
- Hong, Y.S.T., White, P.A., 2009. Hydrological modeling using a dynamic neuro-fuzzy system with on-line and local learning algorithm. *Advances in Water Resources*, 32(1): 110-119.
- Hsia, T.C., 1977. *System identification: Least squares methods*. MA:D.C. Heath, Boston.
- Hsu, K.-L., Gupta, H.V., Sorooshian, S., 1995. Artificial neural network modeling of the rainfall-runoff process. *Water Resources Research*, 31(10): 2517-2530.

- Jacquín, A.P., Shamseldin, A.Y., 2006. Development of rainfall-runoff models using Takagi-Sugeno fuzzy inference systems. *Journal of Hydrology*, 329(1-2): 154-173.
- Jain, A., Kumar, A.M., 2007. Hybrid neural network models for hydrologic time series forecasting. *Applied Soft Computing Journal*, 7(2): 585-592.
- Jain, A., Prasad Indurthy, S.K.V., 2003. Comparative analysis of event-based rainfall-runoff modeling techniques - Deterministic, statistical, and artificial neural networks. *Journal of Hydrologic Engineering*, 8(2): 93-98.
- Jain, A., Sudheer, K.P., Srinivasulu, S., 2004. Identification of physical processes inherent in artificial neural network rainfall runoff models. *Hydrological Processes*, 18(3): 571-581.
- Janacek, G., Swift, L., 1993. *Time series forecasting, simulation, applications*. Ellis Horwood, London.
- Jang, J.-S.R., 1993. ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3): 665-685.
- Karray, F.O., De Silva, C.W., 2004. *Soft computing and intelligent systems design: theory, tools and applications*. Addison Wesley Longman.
- Kasabov, N.K., Song, Q., 2002. DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 10(2): 144-154.
- Keller, J.M., Yager, R.R., Tahani, H., 1992. Neural network implementation of fuzzy logic. *Fuzzy Sets and Systems*, 45(1): 1-12.
- Kohonen, T., 1989. *Self-organization and associative memory*. Springer.
- Krause, P., Boyle, D.P., Bäse, F., 2005. Comparison of different efficiency criteria for hydrological model assessment. *Advances in Geosciences*, 5: 89-97.
- Lee, K.T., Hung, W.C., Meng, C.C., 2008. Deterministic insight into ANN model performance for storm runoff simulation. *Water Resources Management*, 22(1): 67-82.
- Lighthill, M.J., Whitham, G.B., 1955. On kinematic waves II: A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society*, 229(1178): 317-345.
- Lin, C.T., Lee, C.S.G., 1991. Neural-network-based fuzzy logic control and decision system. *IEEE Transactions on Computers*, 40(12): 1320-1336.
- Lin, C.T., Lee, C.S.G., 1996. *Neural fuzzy systems: a neuro-fuzzy synergism to intelligent systems*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

- Lohani, A.K., Goel, N.K., Bhatia, K.K.S., 2011. Comparative study of neural network, fuzzy logic and linear transfer function techniques in daily rainfall-runoff modelling under different input domains. *Hydrological Processes*, 25(2): 175-193.
- Luna, I., Soares, S., Ballini, R., 2007. An adaptive hybrid model for monthly streamflow forecasting. 2007 IEEE International Conference on Fuzzy Systems, FUZZY, London.
- Maguire, L.P., Roche, B., McGinnity, T.M., McDaid, L.J., 1998. Predicting a chaotic time series using a fuzzy neural network. *Information Sciences*, 112(1-4): 125-136.
- Mahabir, C., Hicks, F.E., Fayek, A.R., 2003. Application of fuzzy logic to forecast seasonal runoff. *Hydrological Processes*, 17(18): 3749-3762.
- Maier, H.R., Dandy, G.C., 1997. Determining inputs for neural network models of multivariate time series. *Microcomputers in Civil Engineering*, 12(5): 353-368.
- Mamdani, E.H., Assilian, S., 1975. EXPERIMENT IN LINGUISTIC SYNTHESIS WITH A FUZZY LOGIC CONTROLLER. *International Journal of Man-Machine Studies*, 7(1): 1-13.
- Mariño, P., Pastoriza, V., Santamaría, M., Martínez, E., 2005. Image processing application with a TSK fuzzy model. In: Wang, L., Jin, Y. (Eds.). *Second International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2005, Changsha*, pp. 950-960.
- Masters, T., 1995. *Neural, novel & hybrid algorithms for time series prediction*. John Wiley & Sons, New York.
- MATLAB, 2011b. *MATLAB Fuzzy Logic Toolbox Reference Manual, Version 7.13*
The Math Works Inc. Natick, Mass.
- Minns, A.W., Hall, M.J., 1996. Artificial neural networks as rainfall-runoff models. *Hydrological Sciences Journal*, 41(3): 399-417.
- Mitra, S., Hayashi, Y., 2000. Neuro-fuzzy rule generation: survey in soft computing framework. *IEEE Transactions on Neural Networks*, 11(3): 748-768.
- Monjezi, M., Rezaei, M., 2011. Developing a new fuzzy model to predict burden from rock geomechanical properties. *Expert Systems with Applications*, 38(8): 9266-9273.
- Moon, Y.I., Rajagopalan, B., Lall, U., 1995. Estimation of mutual information using kernel density estimators. *Physical Review E*, 52(3): 2318-2321.
- Mukerji, A., Chatterjee, C., Singh Raghuwanshi, N., 2009. Flood forecasting using ANN, neuro-fuzzy, and neuro-GA models. *Journal of Hydrologic Engineering*, 14(6): 647-652.

- Mutlu, E., Chaubey, I., Hexmoor, H., Bajwa, S.G., 2008. Comparison of artificial neural network models for hydrologic predictions at multiple gauging stations in an agricultural watershed. *Hydrological Processes*, 22(26): 5097-5106.
- Nash, J.E., 1957. The form of the instantaneous unit hydrograph. *The Form of the Instantaneous Unit Hydrograph*, 45(3): 114-121.
- Nash, J.E., Sutcliffe, J.V., 1970. River flow forecasting through conceptual models part I - A discussion of principles. *Journal of Hydrology*, 10(3): 282-290.
- Nauck, D., Klawonn, F., Kruse, R., 1997. *Foundations of neuro-fuzzy systems*. John Wiley & Sons, Inc. New York, NY, USA.
- Nauck, D., Kruse, R., 1995. NEFCLASS - a neuro-fuzzy approach for the classification of data. *Proceedings of the 1995 ACM Symposium on Applied Computing*. ACM, Nashville, TN, USA, pp. 461-465.
- Nayak, P.C., 2010. Explaining internal behavior in a fuzzy if-then rule-based flood-forecasting model. *Journal of Hydrologic Engineering*, 15(1): 20-28.
- Nayak, P.C., Sudheer, K.P., Jain, S.K., 2007. Rainfall-runoff modeling through hybrid intelligent system. *Water Resources Research*, 43(7).
- Nayak, P.C., Sudheer, K.P., Ramasastri, K.S., 2005a. Fuzzy computing based rainfall-runoff model for real time flood forecasting. *Hydrological Processes*, 19(4): 955-968.
- Nayak, P.C., Sudheer, K.P., Rangan, D.M., Ramasastri, K.S., 2004. A neuro-fuzzy computing technique for modeling hydrological time series. *Journal of Hydrology*, 291(1-2): 52-66.
- Nayak, P.C., Sudheer, K.P., Rangan, D.M., Ramasastri, K.S., 2005b. Short-term flood forecasting with a neurofuzzy model. *Water Resources Research*, 41(4): 1-16.
- Noori, R. et al., 2011. Assessment of input variables determination on the SVM model performance using PCA, Gamma test, and forward selection techniques for monthly stream flow prediction. *Journal of Hydrology*, 401(3-4): 177-189.
- Purnomo, M.R.A., Wahab, D.A., Hassan, A., Rahmat, R.A., 2009. A parallel genetic algorithm-based TSK-Fuzzy system for dynamic car-following modeling. *European Journal of Scientific Research*, 28(4): 628-642.
- Qin, X.S., Huang, G.H., He, L., 2010. Development of a cluster-analysis-based distributed hydrologic modeling system, In *proceedings of the 4th International Yellow River Forum (IYRF) on Ecological Civilization and River Ethics*, Zhengzhou, China, October 20-23, 2009, pp. 46-54.

- Quah, K.H., Quek, C., 2006. FITSK: Online local learning with generic fuzzy input Takagi-Sugeno-Kang fuzzy framework for nonlinear system estimation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(1): 166-178.
- Quek, C., Zhou, R.W., 1999. POPFNN-AARS(S): A pseudo outer-product based fuzzy neural network. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 29(6): 859-870.
- Rajurkar, M.P., Kothiyari, U.C., Chaube, U.C., 2002. Artificial neural networks for daily rainfall-runoff modelling. *Hydrological Sciences Journal*, 47(6): 865-878.
- Remesan, R., Shamim, M.A., Han, D., Mathew, J., 2009. Runoff prediction using an integrated hybrid modelling scheme. *Journal of Hydrology*, 372(1-4): 48-60.
- Riad, S., Mania, J., Bouchaou, L., Najjar, Y., 2004. Predicting catchment flow in a semi-arid region via an artificial neural network technique. *Hydrological Processes*, 18(13): 2387-2393.
- Rissanen, J., 1978. Modeling by shortest data description. *Automatica*, 14(5): 465-471.
- Sajikumar, N., Thandaveswara, B.S., 1999. A non-linear rainfall-runoff model using an artificial neural network. *Journal of Hydrology*, 216(1-2): 32-55.
- Shamseldin, A.Y., 1997. Application of a neural network technique to rainfall-runoff modelling. *Journal of Hydrology*, 199(3-4): 272-294.
- Shamseldin, A.Y., 2010. Artificial neural network model for river flow forecasting in a developing country. *Journal of Hydroinformatics*, 12(1): 22-35.
- Shamseldin, A.Y., Nasr, A.E., O'Connor, K.M., 2002. Comparison of different forms of the multi-layer feed-forward neural network method used for river flow forecasting. *Hydrology and Earth System Sciences*, 6(4): 671-684.
- Shamseldin, A.Y., O'Connor, K.M., Liang, G.C., 1997. Methods for combining the outputs of different rainfall-runoff models. *Journal of Hydrology*, 197(1-4): 203-229.
- Shim, K.C., Fontane, D.G., Labadie, J.W., 2002. Spatial decision support system for integrated river basin flood control. *Journal of Water Resources Planning and Management*, 128(3): 190-201.
- Singh, A., Quek, C., Cho, S.Y., 2008. DCT-Yager FNN: A novel Yager-based fuzzy neural network with the discrete clustering technique. *IEEE Transactions on Neural Networks*, 19(4): 625-644.
- Singh, V.P., 1988. *Hydrologic systems. Rainfall-runoff modeling*, vol I. New Jersey, Prentice Hall, 1, 988 pp.

- Snyder, F.F., 1938. Synthetic Unit Hydrographs. Trans. Amer. Geophysical Union, 19:447.
- Sohail, A., Watanabe, K., Takeuchi, S., 2008. Runoff analysis for a small watershed of tono area Japan by back propagation artificial neural network with seasonal data. Water Resources Management, 22(1): 1-22.
- Srinivasulu, S., Jain, A., 2006. A comparative analysis of training methods for artificial neural network rainfall-runoff models. Applied Soft Computing Journal, 6(3): 295-306.
- Sudheer, K.P., 2005. Knowledge extraction from trained neural network river flow models. Journal of Hydrologic Engineering, 10(4): 264-269.
- Sudheer, K.P., Gosain, A.K., Ramasastri, K.S., 2002. A data-driven algorithm for constructing artificial neural network rainfall-runoff models. Hydrological Processes, 16(6): 1325-1330.
- Sudheer, K.P., Jain, A., 2004. Explaining the internal behaviour of artificial neural network river flow models. Hydrological Processes, 18(4): 833-844.
- Takagi, T., Sugeno, M., 1984. DERIVATION OF FUZZY CONTROL RULES FROM HUMAN OPERATOR'S CONTROL ACTIONS, IFAC Proceedings Series. Fuzzy Information, Knowledge Representation and Decision Analysis, Proceedings of the IFAC Symposium. IFAC by Pergamon Press, Marseille, Fr, pp. 55-60.
- 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.
- Tan, S.B.K., Chua, L.H.C., Shuy, E.B., Lo, E.Y.M., Lim, L.M., 2008. Performances of rainfall-runoff models calibrated over single and continuous storm flow events. Journal of Hydrologic Engineering, 13(7): 597-607.
- Tan, S.B.K., Lo, E.Y.M., Shuy, E.B., Chye Chua, L.H., Lim, W.H., 2009. Generation of total runoff hydrographs using a method derived from a digital filter algorithm. Journal of Hydrologic Engineering, 14(1): 101-106.
- Tang, A.M., Quek, C., Ng, G.S., 2005. GA-TSKfnn: Parameters tuning of fuzzy neural network using genetic algorithms. Expert Systems with Applications, 29(4): 769-781.
- Tayfur, G., Singh, V.P., 2006. ANN and fuzzy logic models for simulating event-based rainfall-runoff. Journal of Hydraulic Engineering, 132(12): 1321-1330.
- Thevaril, J., Kwan, H.K., 2005. Speech enhancement using adaptive neuro-fuzzy filtering. 2005 International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2005, Hong Kong, pp. 753-756.

- Tokar, A.S., Johnson, P.A., 1999. Rainfall-runoff modeling using Artificial Neural Networks. *Journal of Hydrologic Engineering*, 4(3): 232-239.
- Tokar, A.S., Markus, M., 2000. Precipitation-runoff modeling using artificial neural networks and conceptual models. *Journal of Hydrologic Engineering*, 5(2): 156-161.
- Tung, W.L., Quek, C., 2002. GenSoFNN: A generic self-organizing fuzzy neural network. *IEEE Transactions on Neural Networks*, 13(5): 1075-1086.
- Turksen, I.B., Zhong, Z., 1988. Approximate analogical reasoning approach based on similarity measures. *IEEE Transactions on Systems, Man and Cybernetics*, 18(6): 1049-1056.
- Turksen, I.B., Zhong, Z., 1990. An approximate analogical reasoning schema based on similarity measures and interval-valued fuzzy sets. *Fuzzy Sets and Systems*, 34(3): 323-346.
- Van Ooyen, A., Nienhuis, B., 1992. Improving the convergence of the back-propagation algorithm. *Neural Networks*, 5(3): 465-471.
- Vernieuwe, H. et al., 2005. Comparison of data-driven Takagi-Sugeno models of rainfall-discharge dynamics. *Journal of Hydrology*, 302(1-4): 173-186.
- Wang, D., Quek, C., Ng, G.S., 2004. Novel Self-Organizing Takagi Sugeno Kang Fuzzy Neural Networks based on ART-like clustering. *Neural Processing Letters*, 20(1): 39-51.
- Werbos, P., 1988. Backpropagation: Past and future. *Proceedings of the 2nd International Conference on Neural Network*.
- Wilby, R.L., Abrahart, R.J., Dawson, C.W., 2003. Detection of conceptual model rainfall-runoff processes inside an artificial neural network. *Hydrological Sciences Journal*, 48(2): 163-181.
- Wong, T.S.W., 2008. Optimum rainfall interval and manning's roughness coefficient for runoff simulation. *Journal of Hydrologic Engineering*, 13(11): 1097-1102.
- Wu, J.S., Han, J., Annambhotla, S., Bryant, S., 2005. Artificial neural networks for forecasting watershed runoff and stream flows. *Journal of Hydrologic Engineering*, 10(3): 216-222.
- Yager, R.R., 1994. Modeling and formulating fuzzy knowledge bases using neural networks. *Neural Networks*, 7(8): 1273-1283.
- Yen, J., Langari, R., 1999. Fuzzy logic: Intelligence, control, and information. *Fuzzy Logic: Intelligence, Control, and Information*.
- Zadeh, L.A., 1965. Fuzzy sets. *Information and Control*, 8(3): 338-353.

- Zadeh, L.A., 1973. OUTLINE OF A NEW APPROACH TO THE ANALYSIS OF COMPLEX SYSTEMS AND DECISION PROCESSES. IEEE Transactions on Systems, Man and Cybernetics, SMC-3(1): 28-44.
- Zadeh, L.A., 1975a. The concept of a linguistic variable and its application to approximate reasoning-I. Information Sciences, 8(3): 199-249.
- Zadeh, L.A., 1975b. The concept of a linguistic variable and its application to approximate reasoning-II. Information Sciences, 8(4): 301-357.
- Zadeh, L.A., 1975c. The concept of a linguistic variable and its application to approximate reasoning-III. Information Sciences, 9(1): 43-80.
- Zealand, C.M., Burn, D.H., Simonovic, S.P., 1999. Short term streamflow forecasting using artificial neural networks. Journal of Hydrology, 214(1-4): 32-48.
- Zhang, C., Wang, Y., Zhang, L., Zhou, H., 2012. A fuzzy inference method based on association rule analysis with application to river flood forecasting. Water Science and Technology, 66(10): 2090-2098.
- Zhou, R.W., Quek, C., 1996. POPFNN: A pseudo outer-product based fuzzy neural network. Neural Networks, 9(9): 1569-1581.
- Zwicky, R., Carlstein, E., Budescu, D.V., 1987. Measures of similarity among fuzzy concepts: A comparative analysis. International Journal of Approximate Reasoning, 1(2): 221-242.

APPENDIX A

Details of the 10 events of Catchment 1.

Event	Date	Duration (min)	Φ -index (mmh ⁻¹)	Rainfall			Runoff from asphalt plane			Remarks
				Peak intensity (mmh ⁻¹)	Average intensity (mmh ⁻¹)	Total depth (mm)	Peak flow (Ls ⁻¹)	Average flow (Ls ⁻¹)	Total depth (mm)	
1	6 Oct 2002	40	5.83	144.0	33.4	22.27	0.670	0.197	19.00	Double peak
2	14 Oct 2002	80	2.64	144.0	19.2	26.56	0.824	0.125	24.12	Double peak
3	27 Oct 2002	40	9.89	216.0	43.2	28.78	0.973	0.263	25.41	Double peak
4	3 Nov 2002	120	1.87	144.0	17.5	35.00	0.816	0.112	32.30	Multiple peaks
5	13 Nov 2002	30	6.29	144.0	32.6	16.30	0.590	0.185	13.41	Double peak
6	17 Nov 2002	100	2.70	144.0	25.7	42.89	0.760	0.164	39.53	Multiple peaks
7	18 Nov 2002	120	0.21	96.0	8.4	16.81	0.528	0.057	16.40	Single peak
8	22 Nov 2002	90	0.80	120.0	19.0	28.57	0.655	0.126	27.38	Multiple peaks
9	5 Dec 2002	100	1.46	144.0	16.6	27.60	0.883	0.106	25.53	Single peak
10	7 Dec 2002	100	1.74	72.0	11.9	19.88	0.342	0.074	17.72	Multiple peaks

APPENDIX B

Details of the 66 events of Catchment 2.

Events No.	Date	Total Rainfall (mm)	Duration (min)	Direct Runoff Volume (m ³)	Rainfall Intensity (mm/min)	Peak Flow (m ³ /s)	Event Group
1	16-Dec-04	12.2	175	18 673	4.2	5.3	3
2	17-Dec-04	29.9	150	35 329	11.9	14.0	1
3	30-Dec-04	12.9	130	12 886	6.0	5.5	1
4	02-Jan-05	85.6	535	75 882	9.6	10.0	3
5	12-Mar-05	20.8	180	21 562	6.9	5.6	3
6	27-Mar-05	9.6	210	6 848	2.8	2.6	3
7	16-Apr-05	47.2	165	59 776	17.2	21.6	2
8	06-May-05	33.2	150	37 542	13.3	13.6	2
9	08-May-05	59.4	350	68 726	10.2	7.3	3
10	16-May-05	38.8	155	53 300	15.0	19.6	2
11	21-May-05	86.1	175	109 401	29.5	42.1	2
12	24-May-05	23.6	150	22 853	9.4	8.8	2
13	25-May-05	41.6	215	41 523	11.6	12.7	2
14	26-May-05	29.0	195	22 846	8.9	8.8	2
15	29-May-05	55.1	180	103 169	18.4	33.4	2
16	01-Jun-05	23.6	140	24 617	10.1	10.3	3
17	08-Jun-05	13.2	115	14 492	6.9	7.2	1
18	04-Jul-05	41.7	125	39 305	20.0	15.4	2
19	17-Jul-05	15.0	165	18 494	5.4	5.6	2
20	18-Jul-05	35.0	280	36 290	7.5	11.6	3
21	19-Jul-05	39.4	145	34 105	16.3	13.2	1
22	26-Jul-05	51.6	235	64 371	13.2	14.2	3
23	30-Jul-05	34.8	145	40 557	14.4	17.4	1
24	02-Aug-05	10.4	165	13 445	3.8	6.2	2
25	04-Aug-05	67.8	145	141 268	28.1	48.9	2
26	20-Aug-05	23.6	180	17 344	7.9	9.2	1

Events No.	Date	Total Rainfall (mm)	Duration (min)	Direct Runoff Volume (m ³)	Rainfall Intensity (mm/min)	Peak Flow (m ³ /s)	Event Group
27	23-Aug-05	41.6	160	64 677	15.6	22.8	2
28	18-Sep-05	18.9	145	19 564	7.8	7.4	2
29	30-Sep-05	20.3	140	25 233	8.7	9.2	2
30	05-Oct-05	55.1	245	58 382	13.5	12.2	2
31	08-Oct-05	22.1	130	20 772	10.2	7.2	1
32	16-Oct-05	98.2	465	122 746	12.7	34.4	3
33	24-Oct-05	12.2	155	12 753	4.7	5.0	2
34	23-Nov-05	39.4	265	49 182	8.9	7.8	3
35	08-Dec-05	24.3	170	37 145	8.6	13.8	2
36	24-Dec-05	73.4	150	88 876	29.4	33.4	2
37	25-Dec-05	19.8	185	39 343	6.4	9.9	3
38	31-Dec-05	34.3	190	35 145	10.8	14.4	2
39	01-Jan-06	9.6	150	12 064	3.9	4.1	2
40	04-Jan-06	18.5	155	33 575	7.2	15.2	2
41	05-Jan-06	50.0	235	75 312	12.8	27.1	3
42	06-Jan-06	278.3	5700	12 064	2.9	28.2	3
43	11-Jan-06	42.1	170	86 372	14.9	22.7	3
44	17-Jan-06	32.8	165	27 575	11.9	8.9	2
45	21-Jan-06	20.0	315	13 662	3.8	2.7	3
46	09-Feb-06	20.3	130	34 467	9.4	17.9	2
47	16-Feb-06	12.4	140	12 667	5.3	5.4	2
48	17-Feb-06	72.4	165	110 963	26.3	38.5	3
49	20-Mar-06	18.5	175	24 438	6.4	7.7	3
50	06-Apr-06	55.6	185	55 458	18.0	12.2	3
51	08-Apr-06	20.1	140	30 476	8.6	13.1	1
52	14-Apr-06	33.5	220	40 352	9.1	10.2	2
53	15-Apr-06	62.7	150	120 542	25.1	43.7	2
54	28-May-06	23.1	275	28 545	5.0	6.3	2
55	06-Jun-06	37.9	210	71 292	10.8	15.8	3

Events No.	Date	Total Rainfall (mm)	Duration (min)	Direct Runoff Volume (m ³)	Rainfall Intensity (mm/min)	Peak Flow (m ³ /s)	Event Group
56	09-Jun-06	58.4	195	108 998	18.0	32.0	3
57	11-Jun-06	54.8	210	112 756	15.7	28.5	3
58	29-Jul-06	23.4	130	42 349	10.8	17.3	1
59	31-Jul-06	30.0	145	45 827	12.4	20.3	2
60	15-Aug-06	38.0	170	54 456	13.4	17.1	2
61	05-Sep-06	34.0	185	36 350	11.0	11.0	2
62	16-Sep-06	40.4	165	58 812	14.7	12.5	2
63	17-Sep-06	30.9	165	38 411	11.2	12.7	2
64	31-Oct-06	78.5	175	66 206	26.9	14.5	3
65	01-Nov-06	21.1	155	31 084	8.2	11.5	2
66	03-Nov-06	59.4	180	75 194	19.8	20.1	3

LIST OF PUBLICATIONS

Journal Papers:

- 1) Talei, A., Chua, L.H.C., Quek, C., 2010. A novel application of a neuro-fuzzy computational technique in event-based rainfall-runoff modeling. *Expert Systems with Applications*, 37(12): 7456-7468.
- 2) Talei, A., Chua, L.H.C., Wong, T.S.W., 2010. Evaluation of rainfall and discharge inputs used by Adaptive Network-based Fuzzy Inference Systems (ANFIS) in rainfall-runoff modeling. *Journal of Hydrology*, 391(3-4): 248-262.
- 3) Talei, A., Chua, L.H.C., Quek, C., 2012. Influence of lag time on event-based rainfall-runoff modelling using the data driven approach. *Journal of Hydrology* 438-439: 223-233.
- 4) Talei, A., Chua, L.H.C., Quek, C., Jansson, P-E., 2013. Runoff forecasting using a Takagi–Sugeno neuro-fuzzy model with online learning. *Journal of Hydrology*, 488: 17-32.

Conference Papers:

- 1) Talei, A., Chua, L.H.C., Quek, C., 2010. The use of adaptive network-based fuzzy inference system (ANFIS) in event-based rainfall-runoff (R-R) modelling., *Proceedings of 9th International Conference on Hydroinformatics, Tianjin, China*, pp. 1391-1398.
- 2) Talei, A., Chua, L.H.C., Quek, C., 2011. Dynamic evolving neural-fuzzy inference system for rainfall-runoff (R-R) modeling, *Proceeding of 34th IAHR World Congress, Brisbane, Australia*, pp. 1514-1521.
- 3) Talei, A., Chua, L.H.C., Quek, C., 2012. Hydrological modeling with a dynamic neural fuzzy inference system. *10th International Conference on Hydroinformatics, 14-18 Jul 2012, Hamburg, Germany*.
- 4) Talei, A., Chua, L.H.C., Quek, C., 2012. Real Time Neural Fuzzy System for Rainfall-Runoff Modeling. *10th International Conference on Hydrosience & Engineering, 4-7 Nov 2012, Orlando, Florida, USA*.