

NOTE TO USERS

This reproduction is the best copy available.

UMI[®]



The Hong Kong Polytechnic University
Department of Civil and Structural Engineering

**Hydrological Predictions Using
Data-Driven Models Coupled with Data
Preprocessing Techniques**

Conglin Wu

A thesis submitted in partial fulfillment
of the requirements for the
degree of Doctor of Philosophy

January 2010

UMI Number: 3448509

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 3448509

Copyright 2011 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it produces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.


Wu Conglin

Abstract

Data-driven models, particularly soft computing models, have become an appropriate alternative to knowledge-driven models in many hydrological prediction scenarios including rainfall, streamflow, and rainfall-runoff. The primary reason is that data-driven models rely solely on previous hydro-meteorological data without directly taking into account the underlying physical process. However, it is inevitable that data-driven models introduce uncertainty to the forecasting as a result of over-simplified assumption, inappropriate training data, model inputs, model configuration, and even individual experience of modelers.

This thesis makes an endeavor to improve the accuracy of hydrological forecasting in three aspects, model inputs, selection of models, and data-preprocessing techniques. Seven input techniques, namely, linear correlation analysis (LCA), false nearest neighbors, correlation integral, stepwise linear regression, average mutual information, partial mutual information, artificial neural network (ANN) based on multi-objective genetic algorithm, are first examined to select optimal model inputs in each prediction scenario. Representative models, such as K-nearest-neighbors (K-NN) model, dynamic system based model (DSBM), ANN, modular ANN (MANN), and hybrid artificial neural network-support vector regression (ANN-SVR), are then proposed to conduct rainfall and streamflow forecasts. Four data-preprocessing methods including moving average (MA), principal component analysis (PCA), singular spectrum analysis (SSA), and wavelet analysis (WA), are further investigated by integration with the abovementioned forecasting models.

K-NN, ANN, and MANN are used to predict monthly and daily rainfall series with linear regression (LR) as the benchmark. The comparison of seven input techniques indicates that LCA is able to identify model inputs reasonably. In the normal mode (viz., without data preprocessing), MANN performs the best, but the advantage of MANN over ANN is not significant in monthly rainfall series forecasting. Compared with results in the normal mode, the improvement of the model performance generated by SSA is considerable whereas MA or PCA imposes negligible influence. Coupled with SSA, advantages of MANN over other models are quite noticeable,

particularly for daily rainfall forecasting.

ANN, MANN, ANN-SVR, and DSBM are employed to conduct estimates of monthly and daily streamflow series where model inputs only depend on previous flow observations. The best model inputs are also identified by LCA. In the normal mode, the global DSBM model shows close performance to ANN. MANN and ANN-SVR tend to be replaceable by each other and are able to noticeably improve the accuracy of flow predictions, particularly for a non-smooth flow series, when compared to ANN. However, the prediction lag effect can be observed in daily streamflow series forecasting. In data preprocessing mode, both SSA and WA bring significant improvement of model performance, but SSA shows a remarkable superiority over WA.

ANN, MANN, and LR are also used to perform daily rainfall-runoff (R-R) prediction where model inputs consist of previous rainfall and streamflow observations. The best model inputs are also attained by LCA. Irrespective of modes, the advantage of MANN over ANN is not obvious. Compared to models depending solely on previous flow data as inputs, these R-R models make more accurate predictions. However, the improvement tends to mitigate with the increase of forecasting horizons in the normal mode. The situation becomes reverse in the SSA mode where the advantage of the ANN R-R model becomes more significant as the prediction horizon increases.

The findings above focused on results of point prediction, which uses the ANN-SSA R-R model. On the basis of this model, we complement this with the uncertainty estimation based on local errors and clustering (UNEEC) method so as to attain interval prediction of daily rainfall-runoff. The UNEEC method is then compared to the bootstrap method. Results indicate that the UNEEC performs better in locations of low flows whereas the bootstrap method proves to be well suited in locations of high flows.

One of the major contributions of this research is the exploration of a viable modeling technique of coupling data-driven models with SSA. The technique has been tested with hydrological forecasts in rainfall, streamflow, and rainfall-runoff, and predicted results are in good agreement with observations.

Publications

Articles in Journals

Wu, C. L., and K. W. Chau (2010), Rainfall-Runoff Prediction Using Artificial Neural Network Coupled with Singular Spectrum Analysis. *Journal of hydrology* (revised)

Wu, C. L., K. W. Chau, and C. Fan (2010), Prediction of Rainfall Time Series Using Modular Artificial Neural Networks Coupled with Data Preprocessing Techniques. *Journal of hydrology*, 389(1-2), 146-167.

K. W. Chau and Wu, C. L. (2009), A hybrid model coupled with singular spectrum analysis for daily rainfall prediction. *Journal of Hydroinformatics*, 12(4), 458-473.

Wu, C. L., K. W. Chau, and Y.S. Li (2009), Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques. *Water Resources Research*, 45, W08432, doi:10.1029/2007WR006737.

Wu, C.L., K.W., Chau, and Y.S., Li (2009), Methods to improve neural network performance in daily flows prediction. *Journal of Hydrology*, 372(1-4), 80-93.

Wu, C.L., K.W., Chau, and Y.S., Li (2008), River stage prediction based on a distributed support vector regression. *Journal of Hydrology*, 358(1-2), 96-111.

Articles in conference proceedings

C.L. Wu, and K.W., Chau (2010), Using ANN-SSA coupled with UNEEC for uncertainty estimate of daily rainfall-runoff transformation. Proceedings of the second international postgraduate conference on infrastructure and environment, June 1-2, Hongkong, China.

Wu C.L., C., Fan, and K.W., Chau (2009), A hybrid ANN-SVR model coupled with singular spectrum analysis for daily rainfall prediction. Proceedings of the 33rd

International Association of Hydraulic Engineering & Research (IAHR) Congress
Congress–Water Engineering for a Sustainable Environment: Multi-process,
Data-driven modeling, August 9-14, Vancouver, British Columbia, Canada.

Wu, C.L. and K.W., Chau (2008), River flow prediction based on a distributed support vector regression. The Second Faculty Postgraduate Research Conference 2008, January 19, 2008, Hong Kong.

Acknowledgments

I have worked on this project for three years. I would like to acknowledge many people for supporting and helping me. Firstly, I would like to thank my supervisor, Prof. Kwok-wing Chau. I cannot remember how many times I felt extremely frustrated by the obstacles I faced in my research. He always gave me pertinent advice, helped me to solve various problems and encouraged me to keep moving forward. Without his support and help, this project would not have been possible. I would also like to thank my co-supervisor, Prof. Yok-sheng Li for his help in my research.

I would like to give my special thanks to Dr. Celia Fan. She is a friendly and brilliant research fellow. She helped me to clarify many of my research questions. Moreover, she also made enormous contributions to my thesis writing.

I would also like to thank Mr. K.W. Lung, Dr. C. Zeng and other friends for promoting a warm and friendly research atmosphere in our office that continuously cultivate my interest in research studies.

I would further like to thank many anonymous reviewers for their pertinent recommendations during journal manuscript review processes which lead to improvement of my thesis.

I am deeply indebted to my wife, my parents and my daughter. Without their continued support and encouragement, never would I have gone so far.

Finally, I gratefully acknowledge the support of Central Research Grant G-U265 of the Hong Kong Polytechnic University.

Notation

| | |
|---------------------|---|
| ACF | Auto-Correlation Function |
| AMI | Average Mutual Information |
| ANFIS | Adaptive Neural Fuzzy Inference System |
| ANN | Artificial Neural Networks |
| ANN ^{MOGA} | Artificial Neural Networks based on Multi-Objective Genetic Algorithm |
| ANN-SVR | Artificial Neural Networks – Support Vector Regression |
| ARMA | Auto-Regressive Moving Average |
| BNN | Bayesian Neural Networks |
| CC | Cross Correlation |
| CE | Coefficient of Efficiency |
| CI | Correlation Integral |
| CWT | Continuous Wavelet Transform |
| DSBM | Dynamic System Based Method |
| DSBM-G | Global Dynamic System Based Model |
| DSBM-L | Local Dynamic System Based Model |
| DWT | Discrete Wavelet Transform |
| ERM | Empirical Risk Minimization |
| FCM | Fuzzy C-Means |
| FDTF | First Difference Transfer Function |
| FIS | Fuzzy Inference Systems |
| FL | Fuzzy Logic |
| FNN | False Nearest Neighbors |
| GA | Genetic Algorithm |
| GBHM | Geomorphology-Based Hydrology Simulation Model |
| GLUE | Generalize Likelihood Uncertainty Estimation |
| GP | Genetic Programming |
| IHDM | Institute of Hydrology Distributed Model |
| KWA | Kinematic Wave Approximation |
| K-NN | K-Nearest-Neighbor |
| LCA | Linear Correlation Analysis |

| | |
|--------|---|
| L-M | Leverberg-Marquardt |
| LR | Linear Regression |
| MA | Moving Average |
| MANN | Modular Artificial Neural Networks |
| MLP | Multilayer Perceptron |
| MLPNN | Multilayer Perceptron Neural networks |
| MOGA | Multi-objective Genetic Algorithm |
| MSDE | Mean Squared Derivative Error |
| MSE | Mean Squared Error |
| MSLE | Mean Squared Logarithmic Error |
| NNM | Nearest-Neighbor Method |
| PACF | Partial Auto-Correlation Function |
| PCA | Principal Component Analysis |
| PI | Persistence Index |
| PICP | Prediction Interval Coverage Probability |
| PMI | Partial Mutual Information |
| PSO | Particle Swarm Optimization |
| RBF | Radial Basis Function |
| RCs | Reconstructed Components |
| RMSE | Root Mean Squared Error |
| R-R | Rainfall-Runoff |
| SCE-UA | Shuffled Complex Evolution |
| SHE | Systeme Hydrologique Europeen |
| SLR | Stepwise Linear Regression |
| SRM | Structral Risk Minimization |
| SSA | Singular Spectrum Analysis |
| SVD | Singular Value Decomposition |
| SVM | Support Vector Machine |
| SVR | Support Vector Regression |
| TSK | Takagi-Sugeno-Kang |
| UNEEC | Uncertainty Estimation based on local Errors and Clustering |
| VC | Vapnik–Chervonenkis |
| WA | Wavelet Analysis |
| XAJ | Xinanjiang |

Table of contents

| | |
|---|-----|
| CERTIFICATE OF ORIGINALITY | i |
| ABSTRACT | ii |
| PUBLICATIONS..... | iv |
| ACKNOWLEDGMENTS | vi |
| NOTATION | vii |
| TABLE OF CONTENTS | I |
| LIST OF TABLES..... | V |
| LIST OF FIGURES | VII |
| 1 Introduction | 1 |
| 1.1 Background | 1 |
| 1.2 Identification of Problems..... | 4 |
| 1.3 Scope and Aim of Research | 10 |
| 1.4 Research Methodology | 11 |
| 1.5 Overview of Thesis | 12 |
| 2 Literature Review..... | 16 |
| 2.1 K-nearest-neighbors method | 16 |
| 2.2 Chaos theory-based method | 19 |
| 2.3 Artificial neural networks..... | 22 |
| 2.4 Fuzzy inference systems | 36 |
| 2.5 Support vector regression..... | 45 |
| 2.6 Genetic programming | 51 |
| 2.7 Comparison of modeling methods | 53 |
| 3 Determination of Model Inputs | 55 |
| 3.1 Linear Correlation Analysis | 55 |

| | |
|---|------------|
| 3.2 Average Mutual Information | 57 |
| 3.3 Partial Mutual Information..... | 60 |
| 3.4 False Nearest Neighbors..... | 62 |
| 3.5 Correlation Integral | 63 |
| 3.6 Stepwise Linear Regression | 64 |
| 3.7 ANN based on multi-objective genetic algorithm..... | 66 |
| 4 Data Preprocessing Methods..... | 68 |
| 4.1 Moving average..... | 68 |
| 4.2 Principal Component Analysis | 69 |
| 4.3 Singular Spectrum Analysis | 71 |
| 4.4 Wavelet Analysis | 77 |
| 5 Potential Forecasting Models | 80 |
| 5.1 Artificial Neural Networks | 80 |
| 5.2 Support Vector Regression | 81 |
| 5.3 Modular Models | 83 |
| 5.4 Models Coupled with Data Preprocessing | 86 |
| 6 Rainfall Forecasting | 89 |
| 6.1 Introduction | 89 |
| 6.2 Study Area and Data..... | 92 |
| 6.3 Applications of Models | 95 |
| 6.4 Results and Discussions | 111 |
| 6.5 Summary | 126 |
| 7 Streamflow Forecasting | 128 |
| 7.1 Introduction | 128 |
| 7.2 Study Area and Data..... | 131 |
| 7.3 Applications of Models | 135 |
| 7.4 Results and Analysis..... | 151 |

| | |
|---|------------|
| 7.5 Summary | 163 |
| 8 Rainfall-Runoff Forecasting..... | 164 |
| 8.1 Introduction..... | 164 |
| 8.2 Case Studies | 168 |
| 8.3 Applications of Models | 170 |
| 8.4 Results and Discussions | 179 |
| 8.5 Summary | 191 |
| 9 Uncertainty Forecasting | 193 |
| 9.1 Introduction..... | 193 |
| 9.2 Methodology | 195 |
| 9.3 Results and Discussions | 201 |
| 9.4 Summary | 209 |
| 10 Summary and Future Work..... | 211 |
| 10.1 Summary | 211 |
| 10.2 Future Work | 213 |
| APPENDIX A K-NEAREST-NEIGHBORS METHOD..... | 215 |
| APPENDIX B NAIVE MODEL | 216 |
| APPENDIX C PRINCIPAL COMPONENT REGRESSION..... | 217 |
| References | 219 |

List of Tables

| | |
|--|-----|
| Table 2.1 Brief summary on pros and cons of data-driven models..... | 54 |
| Table 6.1 Pertinent information for four watersheds and the rainfall data..... | 95 |
| Table 6.2 Comparison of methods to determine mode inputs using ANN model..... | 98 |
| Table 6.3 Performance comparison of ANN with different data-transformed methods | 104 |
| Table 6.4 Model performances at three forecasting horizons under normal mode.. | 112 |
| Table 6.5 Model performances of ANN-MA using the Wuxi data | 115 |
| Table 6.6 Multiple-step predictions for Wuxi and India series using LR, K-NN, and ANN with PCA1 | 116 |
| Table 6.7 Multiple-step predictions for Wuxi and India series using LR, K-NN, and ANN with PCA2 | 117 |
| Table 6.8 Optimal p RCs for model inputs at various forecasting horizons | 119 |
| Table 6.9 Model performances at three forecasting horizons using MANN and ANN with the SSA2 | 120 |
| Table 6.10 RMSE of LR coupled with SSA2 using various L | 126 |
| Table 7.1 Pertinent information for four watersheds and the streamflow data | 134 |
| Table 7.2 Comparison of methods to determine mode inputs using ANN model.... | 137 |
| Table 7.3 Calibrated triplet parameters (C, ε, σ) in MSVR with normal mode..... | 144 |
| Table 7.4 Model performances at three forecasting horizons in normal mode..... | 152 |
| Table 7.5 Optimal p RCs (or DWCs) for ANN's input at various prediction horizons | 156 |
| Table 7.6 Model performances at three forecasting horizons in scheme A | 156 |
| Table 7.7 Comparison of ANN's performance coupled with SSA (or WA) in schemes of A and B | 160 |
| Table 8.1 Statistical information on rainfall and streamflow data | 170 |
| Table 8.2 Comparison of methods to determine mode inputs using ANN..... | 174 |
| Table 8.3 Model structure of static multi-step forecasting..... | 176 |
| Table 8.4 Optimal p RCs for rainfall and streamflow variables at various forecasting horizons | 178 |

| | |
|--|-----|
| Table 8.5 R-R model performances at three forecasting horizons under the normal mode | 179 |
| Table 8.6 Performances of ANN and MANN in two types of input variables..... | 181 |
| Table 8.7 Performances of R-R models in the SSA mode | 183 |
| Table 8.8 Performances of ANN-SSA and MANN-SSA using two types of input variables | 185 |
| Table 9.1 Sensitivity of uncertainty evaluation indices to number of clusters..... | 203 |
| Table 9.2 Relevant information of ANN models for PIs..... | 205 |
| Table 9.3 Comparison of UNEEC and Bootstrap | 207 |
| Table 9.4 Analysis of flow compositions | 207 |

List of Figures

| | |
|--|-----|
| Figure 1.1 Learning/training of data-driven modeling..... | 5 |
| Figure 1.2 Plot of observed against forecasted discharges showing the timing error.. | 9 |
| Figure 1.3 Structure of this thesis | 15 |
| Figure 2.1 A feed-forward ANN structure | 24 |
| Figure 2.2 Activation of a single neuron..... | 24 |
| Figure 2.3 TSK fuzzy inference system | 39 |
| Figure 2.4 Nonlinear SVR with Vapnik's ε -insensitive loss function..... | 47 |
| Figure 2.5 Configuration of SVR..... | 47 |
| Figure 2.6 Tree representation of an expression | 52 |
| Figure 4.1 Schematics of (a) decomposition of x_i at level 3 and (b) signal filter..... | 79 |
| Figure 5.1 Degree of membership of daily rainfall in a representative year using the FCM of two clusters (a) original rainfall series and (b) membership grade | 85 |
| Figure 5.2 Flow chart of hard forecasting using a modular model | 86 |
| Figure 6.1 Location of Daning river basin (Map of Chongqing in the left, and "Wuxi" watershed in the dashed box) | 93 |
| Figure 6.2 Rainfall series of (a) Wuxi, (b) India, (c) Zhenwan, and (d) Zhongxian.. | 94 |
| Figure 6.3 Implementation framework of rainfall forecasting models with/without data preprocessing | 96 |
| Figure 6.4 Plots of m against d_2 for (a) Wuxi ($\tau =4$), (b) India ($\tau =4$), (c) Zhenwan ($\tau =4$), and (d) Zhongxian ($\tau =3$). | 99 |
| Figure 6.5 Plots of ACF and PACF of the rainfall series with the 95% confidence bounds (the dashed lines), (a) and (c) for Wuxi, and (b) and (d) for Zhenwan..... | 100 |
| Figure 6.6 Plots of AMI for (a) Wuxi, (b) India, (c) Zhenwan, and (d) Zhongxian. | 100 |
| Figure 6.7 Plots of FNNP (in the log scale) with m for India rainfall series with $\tau =1$ and R_{tol} varying from 10 to 30..... | 101 |
| Figure 6.8 Check of robustness of K in KNN for (a) Wuxi, (b) India, (c) Zhenwan, and (d) Zhongxian..... | 102 |

| | |
|--|-----|
| Figure 6.9 Singular Spectrum as a function of lag using various window lengths L for (a) Wuxi, (b) India, (c) Zhenwan, and (d) Zhongxian | 106 |
| Figure 6.10 Sensitivity analysis of singular Spectrum on varied τ for (a) Wuxi, (b) India, (c) Zhenwan, and (d) Zhongxian..... | 107 |
| Figure 6.11 Reconstructed components (RCs) and the original series of Zhenwan | 108 |
| Figure 6.12 Plots of CCF between each RC and the raw rainfall data for Zhenwan | 109 |
| Figure 6.13 Supervised procedure for a forecasting model coupled with SSA | 110 |
| Figure 6.14 Scatter plots and hyetographs of one-step-ahead forecast using ANN and MANN for Wuxi ((a) and (c) from ANN, and (b) and (d) from MANN) | 113 |
| Figure 6.15 Scatter plots and hyetographs of one-step-ahead forecast using ANN and MANN for India ((a) and (c) from ANN, and (b) and (d) from MANN) | 114 |
| Figure 6.16 CCFs at three forecast horizons for various lags in time of observed and forecasted rainfall of ANN and MANN: Wuxi (left column) and India (right column)..... | 114 |
| Figure 6.17 Performance of ANN with SSA1 in terms of RMSE as a function of p ($\leq L$) selected RCs at various prediction horizons: (a) Wuxi and (b) India | 118 |
| Figure 6.18 Scatter plots and hyetographs of one-step-ahead forecast using ANN and MANN with SSA2 for Wuxi ((a) and (c) from ANN, and (b) and (d) from MANN) | 121 |
| Figure 6.19 Scatter plots and hyetographs of one-step-ahead forecast using ANN and MANN with SSA2 for India ((a) and (c) from ANN, and (b) and (d) from MANN) | 121 |
| Figure 6.20 CCFs at three forecast horizons using ANN and MANN with Wuxi and India ((a) and (c) for Wuxi and (b) and (d) for India) | 122 |
| Figure 6.21 Histograms of absolute prediction errors of LR, K-NN, ANN, and MANN with the normal mode and SSA2 mode using the Wuxi data..... | 123 |
| Figure 6.22 Histograms of absolute prediction errors of LR, K-NN, ANN, and MANN with the normal mode and SSA2 mode using the India data..... | 123 |
| Figure 6.23 Hyetographs (detail; from 900 th to 950 th) of Wuxi: (1) the raw rainfall series and (2) reconstructed rainfall series for one-step prediction of ANN. | 124 |
| Figure 6.24 CCF between inputs and output for ANN and ANN-SSA2 at three | |

| | |
|---|-----|
| prediction horizons using the Wuxi data..... | 125 |
| Figure 7.1 Map of the Lushui watershed | 133 |
| Figure 7.2 Streamflow series of (a) Xiangjiaba, (b) Wuxi, (c) Danjiangkou, and (d) Chongyang | 133 |
| Figure 7.3 Implementation framework of streamflow forecasting models with/without data preprocessing | 135 |
| Figure 7.4 Plots of m against d_2 for (a) Xiangjiaba ($\tau = 3$), (b) Wuxi ($\tau = 7$), (c) Danjiangkou ($\tau = 3$), and (d) Chongyang ($\tau = 7$)..... | 138 |
| Figure 7.5 Plots of ACF and PACF of the flow series with the 95% confidence bounds (the dashed lines), (a) and (c) for Wuxi, and (b) and (d) for Chongyang | 139 |
| Figure 7.6 Plots of AMI for (a) Xiangjiaba, (b) Wuxi, (c) Danjiangkou, and (d) Chongyang | 139 |
| Figure 7.7 Plots of FNNP (in the log scale) with m for all four stream series..... | 141 |
| Figure 7.8 Flow chart of the ANN-SVR model | 143 |
| Figure 7.9 Singular Spectrum as a function of lag using varied window lengths L from 3 to 10 for (a) Xiangjiaba, (b) Wuxi, (c) Danjiangkou, and (d) Chongyang | 145 |
| Figure 7.10 Sensitivity analysis of singular Spectrum on varied τ for (a) Xiangjiaba, (b) Wuxi, (c) Danjiangkou, and (d) Chongyang | 146 |
| Figure 7.11 RCs and original streamflow series of Wuxi | 146 |
| Figure 7.12 (a) Local wavelet power spectrum using the Morlet wavelet function; (b) global wavelet power spectrum from (a) by averaging in time | 148 |
| Figure 7.13 DWCs and original streamflow series of Wuxi..... | 149 |
| Figure 7.14 Plots of CCF between each DWC and the raw data for Wuxi..... | 149 |
| Figure 7.15 Hydrographs and scatter plots of one-step-ahead forecast for Danjiangkou using (a) ANN, (b) MANN, and (c) ANN-SVR..... | 153 |
| Figure 7.16 Hydrographs and scatter plots of one-step-ahead forecast from (a) ANN, (b) MANN, and (c) ANN-SVR using the Wuxi data | 154 |
| Figure 7.17 CCFs at three forecasting horizons using ANN, MANN, and ANN-SVR (The left column for Danjiangkou, and the right column for Wuxi)..... | 155 |
| Figure 7.18 Hydrographs and scatter plots of one-step lead forecast using (a) | |

| | |
|---|-----|
| ANN-SSA-A, (b) ANN-WA-A and (c) MANN-SSA-A for the Wuxi data. | 157 |
| Figure 7.19 CCFs at three forecast horizons using ANN-SSA-A and ANN-WA-A (The left column for Wuxi, and the right column for Chongyang)..... | 157 |
| Figure 7.20 Norm plots of prediction errors (predicted-observed) of four models using the flow series of (a) Danjiangkou and (b) Wuxi. | 158 |
| Figure 7.21 AMI between each input variable and output of ANN using the Xiangjiaba data..... | 159 |
| Figure 7.22 CCFs at three forecast horizons using ANN-SSA-B and ANN-WA-B (The left column for Wuxi, and the right column for Chongyang)..... | 161 |
| Figure 7.23 AMIs between inputs and output of ANN coupled with three modes at three prediction horizons using the Xiangjiaba data | 162 |
| Figure 7.24 AMIs between inputs and output of ANN coupled with three modes at three prediction horizons using the Danjiangkou data | 162 |
| Figure 8.1 Daily rainfall-runoff time series: (a) Wuxi and (b) Chongyang | 169 |
| Figure 8.2 Implementation framework of rainfall-runoff forecasting models with/without data preprocessing..... | 171 |
| Figure 8.3 CCFs between rainfall and flow series with the 95% confidence bounds (the dashed lines): (a) for Wuxi, and (b) for Chongyang. | 174 |
| Figure 8.4 Singular Spectrum as a function of lag using varied window lengths L from 3 to 10: (a) and (c) for Wuxi, and (b) and (d) for Chongyang..... | 177 |
| Figure 8.5 Sensitivity analysis of singular Spectrum on various τ : (a) and (c) for Wuxi and (b) and (d) for Chongyang. | 177 |
| Figure 8.6 Hydrographs (representative details) and scatter plots of one-step-ahead prediction for Wuxi: (a) LR, (b) ANN, and (c) MANN | 180 |
| Figure 8.7 Hydrographs (representative details) and scatter plots of one-step-ahead prediction for Chongyang: (a) LR, (b) ANN, and (c) MANN. | 180 |
| Figure 8.8 Hydrographs for one-step-ahead prediction using ANN with two types of inputs: (a) Wuxi, and (b) Chongyang. | 182 |
| Figure 8.9 Lag analysis of observation and forecasts of ANN with two types of inputs: (a) and (c) for Wuxi, and (b) and (d) for Chongyang..... | 182 |
| Figure 8.10 Hydrographs (representative details) and scatter plots of one-step-ahead prediction for Wuxi: (a) LR-SSA, (b) ANN-SSA, and (c) MANN-SSA . | 184 |

| | |
|---|-----|
| Figure 8.11 Hydrographs (representative details) and scatter plots of one-step-ahead prediction for Chongyang: (a) LR-SSA, (b) ANN-SSA, and (c) MANN-SSA..... | 184 |
| Figure 8.12 Hydrographs for one-step-ahead prediction using ANN-SSA with two types of inputs: (a) Wuxi, and (b) Chongyang..... | 186 |
| Figure 8.13 Lag analysis of observation and forecasts of ANN-SSA with two types of inputs: (a) and (c) for Wuxi, and (b) and (d) for Chongyang..... | 186 |
| Figure 8.14 AMIs between model inputs and output for ANN with two types of inputs using the Wuxi data..... | 188 |
| Figure 8.15 AMIs between model inputs and output for ANN with two types of inputs using the Chongyang data | 189 |
| Figure 8.16 AMIs between model inputs and output for ANN and ANN-SSA in the context of R-R forecasting using the Wuxi data | 190 |
| Figure 8.17 AMIs between model inputs and output for ANN and ANN-SSA in the context of R-R forecasting using the Chongyang data | 191 |
| Figure 9.1 Terminology used in this chapter..... | 196 |
| Figure 9.2 The generalized framework of the UNEEC method..... | 199 |
| Figure 9.3 Original flow, model residual errors, and norm-plots of errors for Wuxi (the left column) and Chongyang (the right column) | 202 |
| Figure 9.4 AMI between raw model inputs and model errors: (a) Wuxi, and (b) Chongyang. | 203 |
| Figure 9.5 AMI between raw model inputs and PI: (a) and (c) for Wuxi, and (b) ad (d) for Chongyang..... | 204 |
| Figure 9.6 95% prediction limits (a and b) and 95% prediction intervals (c) for Wuxi test data using ANN model to predict one-day lead..... | 206 |
| Figure 9.7 95% prediction limits (a and b) and 95% prediction intervals (c) for Chongyang test data using ANN model to predict one-day lead | 206 |
| Figure 9.9 Representative details of 95% by UNEEC (a) and Bootstrap (b) for Chongyang test data using ANN model to predict one-day lead | 209 |

Part 1

Introduction and Review

1 Introduction

1.1 Background

The need to obtain quick and accurate flood forecasting is crucial in operational management and decision making activities. This is to ensure that a suitable warning period is provided for the authorities to implement their protection or evacuation procedures. In general, watershed hydrology models are designed for these purposes by simulating a catchment's response to rainfall. However, simulating the response using mathematical models is far from a trivial task since the hydrologic processes are complex and the influence of geomorphologic properties and climatic factors on them is not well understood. Consequently, the search for accurate, reliable and physically plausible models still remains one of the most challenging topics in hydrology (de Vos and Rientjes, 2007).

Numerous hydrologic models have been reported since the primitive mathematical modeling — the rational method for peak of discharge — was developed by Mulvany (1850). These hydrologic models can be classified according to their internal description of the hydrologic processes as knowledge-driven or data-driven, and, according to the spatial description of watershed processed, as lumped or distributed (Anctil et al, 2003). Knowledge-driven models are physically-based or process-based, which include mechanistic models and conceptual models. A comprehensive review for knowledge-driven models is provided by Singh and Woolhiser (2002). Mechanistic models generally require describing the behavior of the interacting surface and subsurface processes, which are based on the conservation of mass, momentum and energy. These component models entail input of initial and boundary conditions since flow processes are described by differential equations. In models, parameter values are directly related to catchment characteristics. Examples of mechanistic models are the Systeme Hydrologique Europeen (SHE) (Abbott et al., 1986a, b), Institute of Hydrology Distributed Model (IHDM) (Beven, et al., 1987;

Calver and Wood, 1995), and Geomorphology-Based Hydrology Simulation Model (GBHM) (Yang et al., 1998). Conceptual models are a simplified version of mechanistic models. These models use the principle of mass conservation in conjunction with simplified descriptions of the momentum and energy equations. They use storage elements as the main building components. These storages (also termed buckets or reservoirs) are filled through fluxes, such as, rainfall, infiltration or percolation, and emptied via evapotranspiration, runoff, drainage (Wagener et al., 2004). Compared to mechanistic models, parameters in conceptual models significantly decrease so that the models can be operated easily. Popular examples are the Xinanjiang model (XAJ) (Zhao, 1992), TOPMODEL (Beven, 1995), and the HBV model (Bergstrom, 1995; Lindstrom et al, 1997).

Data-driven models, on the contrary, capture the mapping between input (e.g., rainfall, evaporation, temperature, etc.) and output (i.e. runoff) variables without directly considering the physical laws that underlie the rainfall-runoff process. They are purely based on the information retrieved from the hydro-meteorological data and do not adopt any prior knowledge about catchment behavior and flow processes, hence and also termed black box. Data-driven models can be further divided into two types (Jain and Srinivasulu, 2004): (i) models with predefined structures but unknown parameters (also termed model-based), and (ii) models with unknown structures and unknown parameters. The latter takes advantage of the available time series to derive both the model structure and the corresponding parameter values. The majority of the first type is originated from statistics. Representative examples include simple linear and nonlinear regression models (Box and Jenkins, 1976; Salas et al., 1985), constrained linear systems models (Natale and Todini, 1976), linear perturbation models (Nash and Barsi, 1983), and transfer function models (Kachroo, 1992a; Kachroo, 1992b; Liang et al., 1992). These approaches are based on the assumption that studied time series is a stochastic, normal, and time-invariant process. The second type includes non-parametric methods and computational intelligence (CI) methods, such as K-nearest-neighbor (K-NN) methods (Karlsson and Yakowitz, 1987a; Solomatine et al, 2008), dynamic system based methods (DSBM) (Jayawardena and Lai, 1994), artificial neural networks (ANNs) (Hsu et al., 1995; Solomatine and Dulal, 2003; Khan and Coulibaly, 2006), fuzzy inference systems

estimation (GLUE) for uncertainty estimation of knowledge-driven models, which is quite computationally intensive.

Inversely, the development of data-driven models appears easy and quick since calibrated parameters are only related to model structure and model output (i.e. runoff generation) does not follow the hydrologic processes. Also, unavailability of hydrological data poses practical problems for the application of knowledge-driven models owing to the difficulty of data collection or scarcity of hydro-meteorological networks. In this situation, data-driven modeling seems to be the only alternative, where model inputs can be directly derived from previous rainfall and flow data or flow data only. On the other hand, in some practical situations, the main concern is making accurate and timely predictions at specific locations. A simple data-driven model is adequate by identifying a direct mapping between inputs and outputs. In addition, numerous researchers have reported that data-driven models perform comparably to more sophisticated regression models (Jain et al., 1999; Abrahart and See, 2002; Nayak et al., 2005; Chen et al., 2006). Currently, data-driven model are gaining popularity in hydrology community.

1.2 Identification of Problems

The general procedure of data-driven modeling involves analysis of studied problems, data collection, data preprocessing, model selection, model identification, and evaluation of model. The model identification is a central component, which is termed as learning or training. A typical learning of data-driven modeling is shown in Figure 1.1 where the optimal model can be approximated by minimizing the difference between target and predicted values. A wide range of factors, including data cleaning, model inputs, data scaling, data sets, and model configuration, influence the approximation of a data-driven model to the real hydrological system. A detailed review on these factors will be addressed in Chapter 2, from which several pivotal problems can be identified to underlie the current research.

(FIS) (Liong et al., 2000; Jacquin and Shamseldin, 2006), support vector regression (SVR) (Sivapragasam et al., 2001; Yu et al., 2006) and genetic programming (GP) (Koza, 1992; Babovic and Keijzer, 2000).

Knowledge-driven models have the principal advantage that they explicitly capture runoff-generation mechanisms in terms of the underlying physical processes such as interception, infiltration, exfiltration, the lateral distribution of soil moisture and the localized saturation of the shallow subsurface in a spatially-distributed fashion (Arduino et al., 2005). These models make predictions that may be distributed in space so that the effects of partial changes to the catchment, and the predicted spatial dynamics of the processes can be assessed (Beven, 2000). Besides, their model parameters would have direct physical interpretation, and their values might be established by field or laboratory investigations (Kachroo, 1992a). Therefore, knowledge-driven models are currently preferred for watershed management and engineering design by hydrologists and water resources specialists.

However, the calibration remains a chief obstacle in knowledge-driven modeling. Based on the assumption that a set of parameter values exist and provide the optimum simulation of the observations, the aim of calibration is to identify this optimal parameter set in terms of some objective functions. Finding the optimal set is not a simple task. It necessitates a large quantity of hydro-meteorological data and robust optimization technique. The optimization may suffer from varied problems including over-parameterized effects, parameter redundancy effects (existence of insensitive parameters), local optima in the search surface, and large computational demands. Moreover, the existence of the optimal set is truly questioned because it is changeable when the model is calibrated using different calibration periods. The phenomenon implies that any of these model classes may not reflect the true rainfall-runoff relationship. Therefore, convergence to the optimal model may be impossible no matter how large the database becomes. To overcome this difficulty, the idea of equifinality is introduced to hydrologic forecasts (Beven, 1993) where multiple possibilities for simulations from acceptable simulators in some sense are advocated with a rejection of the concept of the optimal model. Beven and Binley (1992) expanded this idea to the frame of generalized likelihood uncertainty

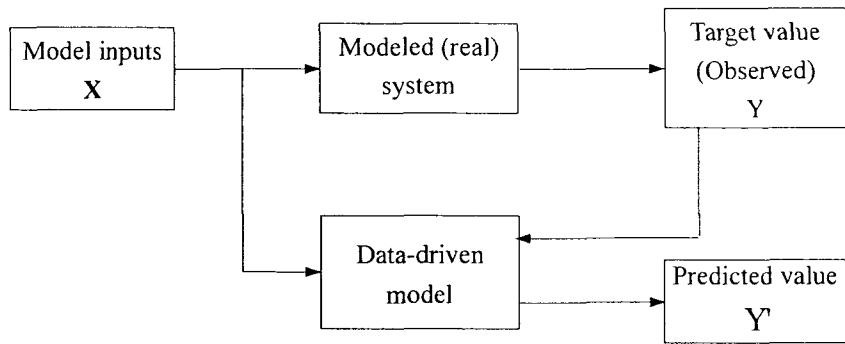


Figure 1.1 Learning/training of data-driven modeling (adopted from Mitchell, 1997)

1.2.1 Model inputs and parameters optimization

Data-driven models aim primarily at establishing functional relationships between input and output variables without defining a known functional relationship (except for classical regression models). The model identification (viz. determining inputs, structure, and parameters) depends almost completely on training data. Determining candidate input variables is a fundamental and crucial task. Without appropriate model inputs, any advanced model structure and parameter optimization algorithm can be in vain for model identification because an identified model is likely to misrepresent the real system. There are many approaches pertaining to model input selection (see Chapter 2). They are either model free or model dependent as well as either linear or nonlinear. However, most of researchers have employed only one or two of these methods. There is limited research related to comparison of these methods. Sudheer et al. (2002) compared linear correlation analysis (LCA) with trial and error approach to prove the feasibility of the former in finding the optimal model inputs.

Model parameters can be treated as an important internal factor in influencing the model performance. These parameters control robustness and generalization of models. The best result is commonly acquired by maximizing a likelihood function of the parameters or equivalently minimizing an error function (Zhang and Govindaraju, 2000). However, the mapping space formed by hydrologic data can be fragmented or discontinuous, which means that a large number of local minima points exist in the whole search regions. Global competitive algorithms such as

genetic algorithm (GA), particle swarm optimization (PSO) and Shuffled Complex Evolution (SCE-UA) should be taken into consideration instead of classical algorithms such as gradient descent or quasi-Newton methods. Certainly, the issue of parameters optimization is very case-specific. Therefore, an examination of optimization algorithms may contribute to the determination of the optimal model.

1.2.2 Modular modeling method

The determination of appropriate modeling approach and adequate model structure still remains great concern by researchers. Table 2.1 reveals that no modeling method prevails over others because each modeling technique exhibits merits and drawbacks. For example, Laio et al. (2003) carried out a comparison of the chaos theory-based approach and ANN for flood predictions and discovered that the former performed slightly better at short forecast leads whereas the situation was reverse for longer leads. Sivakumar et al. (2002) found that the performance of the chaos theory-based approach was better than that of ANN in short-term river flow prediction. In contrast, Yu et al. (2004) found that the chaos theory-based approach performed the worse compared to autoregressive integrated moving average (ARIMA) and SVR for daily runoff forecasting. The FIS technique is not always superior to those of far simpler methods, such as autoregressive moving average (ARMA) models (See and Openshaw 2000) or multiple linear regressions (Jacquin and Shamseldin 2004). These findings confirm that such perfect model that will perform well on all catchments, at all times, and in all circumstances, does not exist (O'Connor, 2005).

Consequently, ensemble models and hybrid (or modular) models were recently proposed for the purpose of combining different modeling methods. An example of ensemble prediction was reported by See and Openshaw (1999) where several ANNs were combined by a fuzzy logic method to form a single forecasting system. The combination of chaos theory and SVR can be found in Sivapragasam et al. (2001). A well-known integration of ANN and FIS, termed adaptive neuro-fuzzy inference system (ANFIS) model, was designed by Jang (1993), which has been widely used by hydrological researchers. Modular model for streamflow prediction is derived

from a well-recognized fact that mechanisms governing different flow events (such as high flows or low flows) are characteristically different. For example, the high flows depend primarily upon the outburst of a heavy storm in the immediate past and geomorphologic properties (such as topology, vegetation, soil type and land use). The low flows, on the other hand, are affected primarily by releases from ground water storage, which is a function of the distribution and infiltration characteristics of the soils, the hydraulic characteristics and the extent of the aquifers, the rate, frequency and amount of recharge (Sivapragasam and Lioung, 2005). Therefore, a global data-driven model may not be capable of describing all hydrological situations adequately. As a consequence, modular models are proposed where hydrological situations are first of all partitioned into several classifications and separate models (also termed local or expert model) are then established for each of them (Solomatine and Ostfeld, 2008). Depending on the soft or crisp split of training data, different modular models exist. Soft split means the dataset can be overlapped and the overall forecasting output is the weighted-average of each local model (Zhang and Govindaraju, 2000; Shrestha and Solomatine, 2006; Wu et al., 2008). On the contrary, there is no overlap of data in the crisp split and the final forecasting output is explicitly from a particular local model (Corzo and Solomatine, 2007; Jain and Srinivasulu, 2006; See and Openshaw, 2000; Sivapragasam and Lioung, 2005; Solomatine and Xue, 2004). ANN (or similar techniques) is unable to extrapolate beyond the range of the data used for training. Otherwise, poor forecasts or predictions can be expected when a new input data is outside the range of those used for training. As far as SVR is concerned, this problem was indeed noticed by Wu et al. (2008). Therefore, modular SVR derived from crisp data split should be proposed to avoid the poor extrapolation. However, Wang et al. (2006b) compare predictions from hard (threshold values classification) and soft (fuzzy c-means clustering) methods for validation data using the threshold value-based (in essence, crisp-based split) periodic ANN. Results showed that soft partitioning method provides a slightly better prediction than the hard partitioning method. This is worthwhile to have further investigation. In the meantime, Table 2.1 also shows that there is still vast room to develop various hybrid models because some modeling methods are mutually complementary. For example, ANN can deal with large samples with a fast training speed but easily suffer from local minima. SVR and fuzzy logic (FL) are

adept in operating small training samples with good generalization and are more transparent. Therefore, developing a hybrid model of ANN combined with SVR or FIS, which takes advantage of their virtues, seems to be feasible. The hybrid model can be expected to achieve better forecast performance than an individual model.

1.2.3 Timing error

Timing error (or phase shift errors) between observed and predicted values appears to be a common problem. Many studies have indicated that phase shift errors indeed occurred in some data-driven models (Minns, 1998; Campolo et al., 1999; Dawson and Wilby, 1999; Zealand et al., 1999; Thirumalaiah and Deo, 2000; Gaume and Gosset, 2003; Jain and Srinivasulu, 2004; de Vos and Rientjes, 2005; Yu and Chen, 2005; Mutti and Chau, 2006; de Vos and Rientjes, 2007). Figure 1.2 highlights the timing errors, where the forecasts are shifted forward. Only a few studies have addressed this problem. For example, Minns (1998) observed a phase shift error in forecasting outputs when antecedent discharge values were the only inputs used to forecast present discharge. However, models developed using discharge and rainfall inputs were not observed to exhibit phase shift errors. Varoonchotikul (2003) proposed a method that could be used to suppress phase shift error in an ANN rainfall-runoff model based on the First Difference Transfer Function (FDTF: Duband *et al.*, 1993). FDTF is developed from the unit hydrograph and provides an initial forecast of future discharge which is then used as another input to the ANN model. The inclusion of this “forward shift operator” was observed to produce accurate results with no phase shift errors. The neural solutions had no rainfall inputs but rainfall records were used to develop the forward shift operators to ensure that rainfall controls were not absent in the model. de Vos and Rientjes (2005) identified two potential sources of timing error: (a) autoregressive components used as input variables; and (b) optimization procedures used as a specific objective function (viz. root mean squared error (RMSE)). It was suggested that autoregressive components dominated the ANN rainfall-runoff model and other important inputs, such as rainfall, had less influence on the model output. Therefore, the authors conducted a series of rainfall-runoff experiments in which different input variables were used to examine the effect of strong autocorrelation on the model outputs in terms of both timing and overall fit. The best result was obtained using a combination of

hydrological state representers: a moving average (MA) over the previous discharge, a moving average over the previous rainfall, and the output of the simple GR4J soil moisture model. MA adjusted the contribution of each input to model output. For example, it tends to decrease the weight of the latest input component to model output, and to increase the weight of the more previous input. From the perspective of optimization procedures, de Vos and Rientjes (2007) developed an ANN rainfall-runoff forecast model with three objective functions, viz. mean squared error (MSE), mean squared logarithmic error (MSLE), and mean squared derivative error (MSDE). The MSDE is an indicator of the fit of the shape of the hydrograph. It especially penalizes noise time series and series with timing error. Abrahart et al. (2007) introduced a “time-shift function” (also termed correction factor) to the objective function of RMSE to correct timing error. The method implied that the lag error was corrected at the expense of an increase RMSE. Evidently, the essence of objective functions-based operations is to seek a compromise between timing error and model performance. Yu and Chen (2005) observed phase shift errors in predicted hydrographs when using grey rainfall-runoff forecasting models for one- to four-hour ahead predictions. They adopted an error prediction model based on a fuzzy rule-based method to update forecasted results so as to mitigate the phase lag.

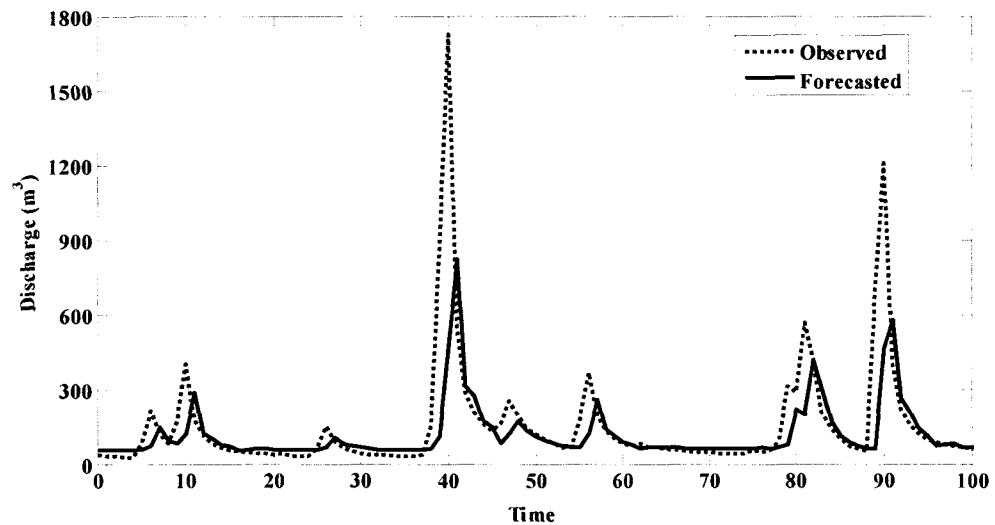


Figure 1.2 Plot of observed against forecasted discharges showing the timing error

Apart from MA, other data preprocessing techniques including principal component analysis (PCA), wavelet analysis (WA) and singular spectrum analysis (SSA), have been successfully applied to the forecasting of water-related issues including rainfall,

discharge and sediment (Sivapragasam et al., 2001; Küçük and Ağaloğlu, 2006; Marques et al., 2006; Hu et al., 2007; Partal and Kişi, 2007; Partal and Cigizoglu, 2008). These studies demonstrated that the performance of model fed by treated data achieved a significant improvement when compared to corresponding benchmark model which was fed by raw data. It can be observed by graphic results in those studies that there were no phase shift errors between predictions and target values although authors did not explicitly mention the issue. It is therefore believed that these data preprocessing techniques may be appropriate for elimination of timing error.

Another important reason for the use of data preprocessing techniques is that a hydrological time series is actually an integration of stochastic (or random) and deterministic components (Salas et al., 1985). Once the stochastic (noise) component is appropriately eliminated, the deterministic component can then be easily modeled. The work of Sivapragasam et al. (2007) demonstrated that short lead predictions can be significantly improved from a short and noisy time series filtered by SSA whereas the superiority gradually disappears for longer leads.

1.3 Scope and Aim of Research

A hydrologic forecasting may include all or some parts of the following three basic elements (Yu and Tseng, 1996): (i) a precipitation forecasting model; (ii) a rainfall-runoff transformation model; and/ or (iii) a river flow routing model. As far as rainfall-runoff transformation is concerned, the observed rainfall may be adequate for runoff estimate because an operational prediction step is often within the response time of a large-size watershed. However, forecasted rainfall is often required for small and medium-sized mountainous basins where expected lead time of the runoff forecast is possibly larger than the response time of the watershed. The scope of this research therefore includes three types of forecasts, namely, rainfall time series forecasting, streamflow time series forecasting, and rainfall-runoff forecasting. The time resolution of the experimental data is on the basis of monthly or daily interval.

The thesis aims at developing robust data-driven forecasting models with the aid of data preprocessing techniques which can make a quick and accurate prediction for rainfall or streamflow. Four main objectives are concerned with the aim:

- To evaluate seven candidate model input methods (namely, linear correlation analysis (LCA), false nearest neighbors (FNN), correlation integral (CI), stepwise linear regression (SLR), average mutual information (AMI), partial mutual information (PMI), and ANN based on multi-objective genetic algorithm (ANN^{MOGA});
- To compare various data-driven models, both single and modular (or hybrid) in the context of monthly and daily rainfall and streamflow predictions;
- To determine the optimal one among four data preprocessing techniques (namely, moving average (MA), principal component analysis (PCA), singular spectrum analysis (SSA), and wavelet analysis (WA)) for the purpose of eliminating timing error;
- To extend point forecasts to interval forecasts (or uncertainty forecast) which make forecasted values with specified confidence limits.

1.4 Research Methodology

This research bases methodology on a general procedure, i.e. analysis of studied problems, modeling, and applications.

The performance of data-driven models in conjunction with data preprocessing techniques is investigated by three types of forecasting experiments, namely, rainfall, streamflow, and rainfall-runoff transformation. To ensure wider application of drawn conclusions, monthly and daily rainfall and streamflow data from varied regions or river basins are examined. Based on aforesaid identified problems, we pay much attention on three aspects in the modeling process: model inputs, data preprocessing, and development of models. Seven candidate methods for determination of model inputs (viz., LCA, FNN, CI, SLR, AMI, PMI, and ANN^{MOGA}), are first compared for selection of optimal model inputs in each prediction scenario. Representative models of K-NN, DSBM, ANN, MANN, and ANN-SVR, are then proposed to conduct

rainfall and streamflow forecasts. Four data preprocessing methods including MA, PCA, SSA, and WA, are further investigated by combining them with proposed forecasting models.

All mentioned models and data preprocessing techniques are not repeatedly investigated in each experiment. In the rainfall experiment, seven model input methods, four rainfall forecasting models of LR, K-NN, ANN and MANN, and three data preprocessing techniques of MA, PCA and SSA, are examined. In the streamflow experiment, seven model input methods, four streamflow forecasting models of ANN, MANN, ANN-SVR and DSBM, and two data preprocessing techniques of SSA and WA, are tested. Moreover, five model input methods excluding FNN and CI, and three models of LR, ANN and MANN combined with SSA, are examined in the experiment of rainfall-runoff transformation.

1.5 Overview of Thesis

The section gives a brief outline of this thesis. The thesis is organized as Figure 1.3. Its contexts principally comprise four parts including ten chapters, each of which is summarized as follows.

Chapter 1 Introduction

This chapter presents the research background, identifies research problems, defines the aim and scope of study, describes research methodology, and outlines chapters in the thesis.

Chapter 2 Literature Review

In this chapter, several popular data-driven modeling approaches, namely, K-NN, chaos theory-based method, ANN, FIS, SVR and GP, are reviewed. Contents include modeling method, issues related to model identification, applications in hydrology, and their merits and drawbacks.

Chapter 3 Determination of Model Inputs

Seven conventionally used methods to determine model inputs are presented. They will be examined by ANN to identify the optimal model inputs for later hydrological prediction experiments.

Chapter 4 Data Preprocessing Methods

This chapter introduces four commonly used approaches of data preprocessing, namely, MA, PCA, SSA, and WA. These methods have potential capabilities to improve the mapping between model inputs and output.

Chapter 5 Potential Forecasting Models

This chapter primarily concentrates on the introductions of ANN and SVR, and modular models derived from them. Moreover, the combination of forecasting models with data preprocessing techniques is also presented.

Chapter 6 Rainfall Forecasting

This chapter describes applications of four data-driven models coupled with SSA to monthly and daily rainfall time series forecasts. K-NN, ANN, and MANN are used to predict monthly and daily rainfall with LR as the benchmark. Comparison of seven input methods indicates that LCA is able to reasonably ascertain model inputs. In the normal mode (viz., without data preprocessing), MANN performs the best, but the advantage of MANN over ANN is not significant in monthly rainfall series forecasting. Compared with results in the normal mode, the improvement of the model performance derived from SSA is considerable whereas that from MA or PCA is almost negligible. Coupled with SSA, MANN displays noticeable advantages over other models, in particular for daily rainfall forecasting. It can also be found that the unsupervised filter method for selection of effective components of SSA is generally better than the supervised method.

Chapter 7 Streamflow Forecasting

The primary objective of this chapter is to investigate the performance of ANN-SVR and carry out a comparison between SSA and WA. ANN, MANN, ANN-SVR, and DSBM are employed to conduct estimates of monthly and daily streamflow where model inputs depend on previous flow observations and the best model inputs are

identified by LCA. In the normal mode, the global DSBM shows a close performance to ANN. MANN and ANN-SVR tend to be alternative to each other and are able to noticeably improve the accuracy of flow predictions, in particular for an irregular flow series, when compared to ANN. However, the prediction lag effect can be observed in daily streamflow series. In the data preprocessing mode, both SSA and WA bring about significant improvement of model performance, but SSA shows a remarkable superiority over WA.

Chapter 8 Rainfall-Runoff Forecasting

Based on the same watersheds for daily streamflow prediction (see Chapter 7), ANN, MANN, and LR are used to perform daily R-R prediction where model inputs consist of previous rainfall and streamflow observations. The best model inputs are also attained by LCA. Regardless of modes, the advantage of MANN over ANN is slight and seems to be sensitive to study cases. Compared to models depending solely on previous flow data as inputs, these R-R models render predictions with better accuracy. The improvement tends to mitigate with the increase of forecasting horizons in the normal mode. However, situation becomes reverse in the SSA mode where the advantage of the ANN R-R model becomes salient with increasing prediction leads.

Chapter 9 Uncertainty Forecasting

Depending on the ANN-SSA R-R model as point prediction model, interval prediction of daily rainfall-runoff is addressed by the method of uncertainty estimation based on local errors and clustering (UNEEC) compared to the bootstrap method. Results indicate that UNEEC performs better for low flows whereas the bootstrap method proves to be better for high flows.

Chapter 10 Summary and Future Work

This chapter presents findings and contributions of this research and provides several issues for further research in this area.

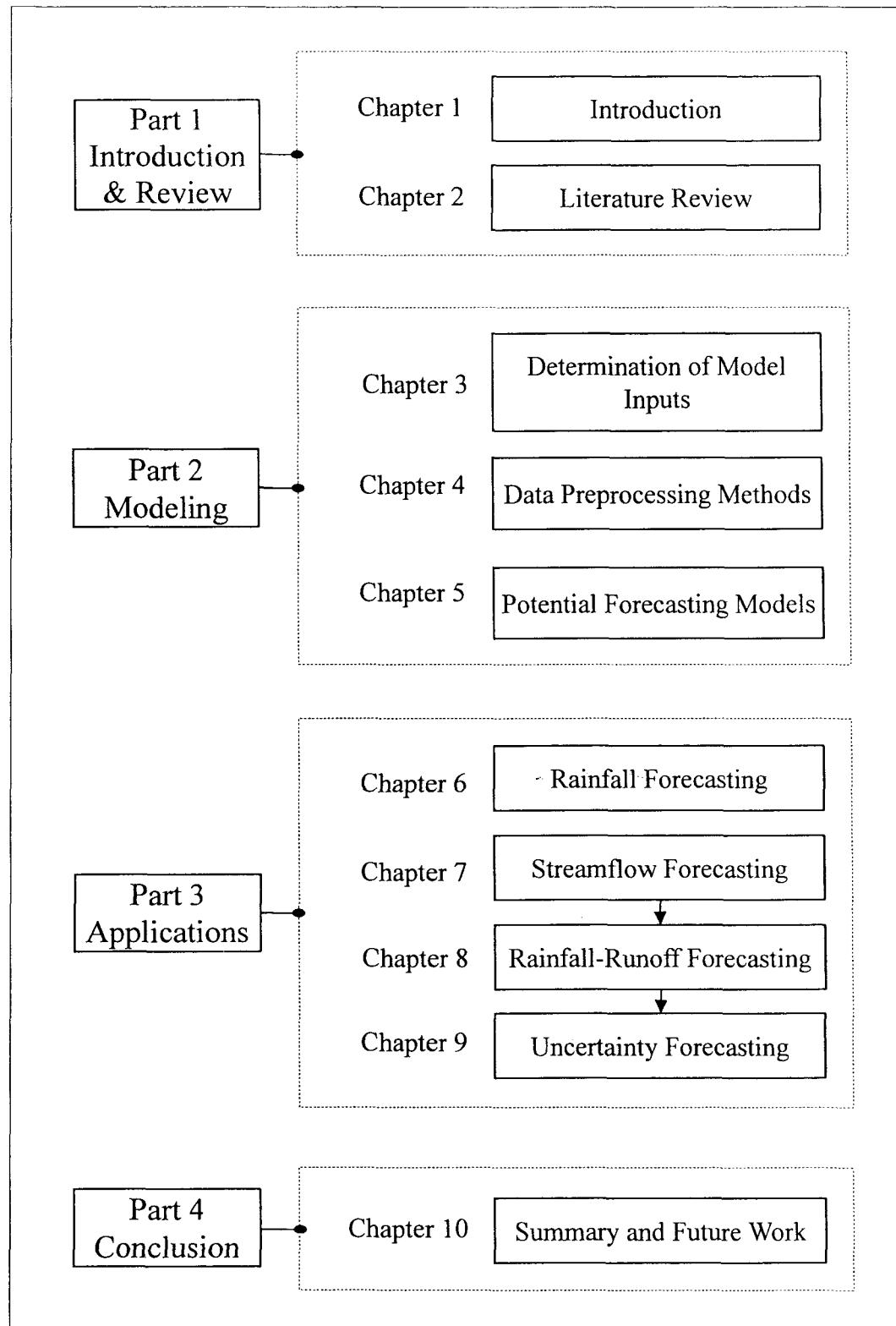


Figure 1.3 Structure of this thesis

2 Literature Review

In this chapter, several popular data-driven modeling approaches are reviewed including K-NN, chaos theory-based method, ANN, FIS, SVR and GP. The reviews of the modeling approaches involve introduction of modeling method, issues related to model identification, applications in hydrology, and their merits and drawbacks.

2.1 K-nearest-neighbors method

2.1.1 Introduction

The K-NN method is based on local approximation which makes use of only nearby observations of the point of estimate. It has its origins as a non-parametric statistical pattern recognition procedure, aiming at distinguishing between different patterns according to chosen criteria (Mack and Rosenblatt, 1979; Aha *et al.*, 1991). Amongst various non-parametric kernel methods, the K-NN method seems more intuitive, but nevertheless possesses powerful statistical properties (Karlsson and Yakowitz, 1987a, b). Yakowitz (1987) and Karlsson and Yakowitz (1987a, b) did considerable work in extending the K-NN method to time-series and forecasting problems, obtaining satisfactory results and constructing a robust theoretical base for the K-NN method.

To begin with, let us describe the K-NN method in the context of rainfall-runoff forecasting. For each time epoch t , let $\mathbf{X}(t) = (p_t, p_{t-1}, \dots; q_t, q_{t-1}, \dots)$ be a feature vector consisting of past rainfall (p) and runoff (q) recordings. Suppose that the runoff $q(t+1)$ at one-step lead is related to the vector $\mathbf{X}(t)$, the available historical data may be summarized into a set of pairs as $\{\mathbf{X}(t), q(t+1) : t = 1, \dots, n\}$, where n stands for the number of instances. The prediction of $q(n+1)$ by the K-NN method is formulated as:

$$\hat{q}(n+1) = \frac{1}{K} \sum_{t \in S(\mathbf{X}, n)} q(t+1) \quad (2.1)$$

where $S(\mathbf{X}, n)$ denotes the set of indices t of the K nearest neighbors to the feature vector $\mathbf{X}(n)$. The meaning of “nearest neighbors” is generally interpreted in a Euclidean sense. Therefore, if i belongs to $S(\mathbf{X}, n)$ and j is not in $S(\mathbf{X}, n)$, $\|\mathbf{X}(n) - \mathbf{X}(i)\| \leq \|\mathbf{X}(n) - \mathbf{X}(j)\|$ according to Euclidean distance. Intuitively speaking, the forecast $\hat{q}(n+1)$ in Eq. (2.1) is the sample average of succeeding runoff of the K nearest neighbors to $\mathbf{X}(n)$. The K-NN algorithm is illustrated by the example of discharge prediction from the work of Karlsson and Yakowitz (1987a) in Appendix A.

When \mathbf{X} comprises different hydro-meteorological data, normalization is usually necessary because it eliminates the units from components or elements of \mathbf{X} and reduces any differences in the range of values among components, for example, rainfall and discharge, with their different units and scales. In order to reflect the relative importance because the more recent measurements in the feature vector generally have a greater weight towards predicted values, the Euclidean distance can be computed as a weighted Euclidean norm, i.e., $\|\mathbf{d}\|_w = \left(\sum_{i=1}^m w_i \times d_i^2 \right)^{1/2}$ where $w = (w_1, w_2, \dots, w_m)$ is a fixed sequence of positive numbers (weights). On the other hand, in order to reflect the relative contribution to prediction value, each of all K neighbors is set to a weight factor ω_i which is based on the Euclidean distance. The prediction model becomes

$$\hat{q}(n+1) = \frac{1}{K} \sum_{t \in S(\mathbf{X}, n)} q(t+1) \cdot \omega_t \quad (2.2)$$

where $\omega_t = \|d_t\|^{-2} / \sum_{i=1}^K \|d_i\|^{-2}$. There are also other approaches to calculate weight factors (e.g., Jayawardena and Lai, 1994; Solomatine et al., 2008).

2.1.2 Issues related to model identification

K-NN does not require the selection of a class of models and the estimation of the model parameters, so that the identification of a specific form of the input/output relationship is not needed. However, two parameters, the number of nearest

neighbors K and the dimension of the feature vector m , need to be determined. The K and m can be obtained by a global optimization method such as GA concurrently or by trial and error separately. The conventionally-used way is by trial and error. For calibration purpose, the training data should be divided into two parts: one is used to construct the nearest-neighbors-predictors (constructing patterns); the other is used to calibrate parameters. The optimal K is achieved by minimizing $J(K) = \sum (q(t+1) - \hat{q}(t+1))^2, t = 1, \dots, n$, where $q(t+1)$ is observed value.

2.1.3 Applications

The intuitiveness and the powerful theoretical basis have made K-NN attractive to forecasters, especially in the hydrologic field, where the method is found successful in applications (Karlsson and Yakowitz, 1987a; Galeati, 1990; Kember and Flower, 1993; Todini, 1996; Toth et al. 2000; Bannayan and Hoogenboom, 2008; Solomatine et al., 2008). Karlsson and Yakowitz (1987a) originally employed the K-NN technique to predict rainfall-runoff transformation in comparison with autoregressive moving average model with auxiliary input (ARMAX) and instantaneous unit hydrology (IUH) forecasters. Results showed that K-NN is similar to ARMAX but better than IUH. Galeati (1990) applied K-NN (with the vectors composed of the lagged rainfall and flow values) to daily discharge forecasting. Shamseldin and O'Connor (1996) used K-NN in adjusting the parameters of the linear perturbation model for river flow forecasting. Toth et al. (2000) compared K-NN, ARMA, and ANN in the context of short-term rainfall forecasting, and results demonstrated that ANN performed the best. Solomatine et al. (2008) investigated the performance of K-NN compared to ANN and M-5 model trees for hourly and daily rainfall-runoff predictions, and results demonstrated that K-NN is comparable with other data-driven models, particularly when coupled with the Gaussian.

2.1.4 Advantages and disadvantages

The learning algorithm in K-NN is quite simple but robust, can tolerate noise and irrelevant attributes, and can represent both probabilistic and overlapping concepts and naturally exploit inter-attribute relationships (Aha et al., 1991). K-NN is able to

permits to identify the instances (hydrological events) in the past, which makes it be more transparent than ANN and easily accepted by decision-makers. It does not detect any input/output mapping function, not even *a posteriori* (whereas the ANNs do), and it has, therefore, no extrapolation ability when presented with an unfamiliar input vector. Thus, in no case can a value higher than the historical discharges be predicted, which severely restricts the plausibility of the K-NN method when used in actual forecasting.

2.2 Chaos theory-based method

2.2.1 Introduction

The chaos theory has been introduced to the field of hydrology where a hydrologic time series is viewed as a chaotic dynamic system with a low-dimensional attractor (e.g., Jayawardena and Lai, 1994; Porporato and Ridolfi, 1997; Sivakumar et al. 2002). The system can be described by a phase-space diagram which is achieved by reconstruction of data series. The most popular reconstruction method is the time delay method, proposed by Takens (1981) for univariate time series and extended to multivariate series by Sauer et al. (1991). Let $\{x_1, x_2, \dots, x_N\}$ stand for a dynamic time series. It can be reconstructed into a series of delay vectors as $\mathbf{Y}_t = \{x_t, x_{t+\tau}, x_{t+2\tau}, \dots, x_{t+(m-1)\tau}\}$, $t = 1, 2, \dots, n$, where $\mathbf{Y}_t \in \mathbb{R}^m$, $n = N - (m-1)\tau$, τ is the delay time as a multiple of the sampling period and m is the embedding dimension. Prediction can be conducted by temporal evolution of the dynamic system. Its temporal evolution for a T -step lead is given as a mapping $\mathbf{Y}(t) \mapsto \mathbf{Y}(t+T)$ (or $\mathbf{Y}_t \mapsto \mathbf{Y}_{t+T}$). The functional relationship between the current state $\mathbf{Y}(t)$ at time t and the predicted state $\mathbf{Y}^F(t+T)$ at time $t+T$ can be written as follows:

$$\mathbf{Y}^F(t+T) = f(\mathbf{Y}(t)) + e_t \quad (2.3)$$

where e_t is a typical noise term. In the form of time series, it can be expressed as

$$[x_{t+T}^F, x_{t+T+\tau}^F, \dots, x_{t+T+(m-2)\tau}^F, x_{t+T+(m-1)\tau}^F] = f([x_t, x_{t+\tau}, \dots, x_{t+(m-2)\tau}, x_{t+(m-1)\tau}]) + e_t. \text{ Therefore,}$$

predicting future trajectory by current trajectory becomes viable once the function $f(\bullet)$ is determined. In practice, the expression is often defined as:

$$x_{t+\tau+(m-1)\tau}^F = f(Y(t)) + e, \quad (2.4)$$

where only the last component in $Y^F(t+\tau)$ is indicated since normally the prediction of this last component is of concern (Laio et al., 2003).

2.2.2 Issues related to model identification

Existence of the attractor in a time series is precondition of the chaos theory-based method. Commonly-used methods of chaos identification include the correlation dimension (D_2), the largest Lyapunov exponent (λ_1) and the Kolmogorov entropy (K_2). A time series can be considered to be chaotic once the following criterion satisfies: the saturation value D_2 of the correlation dimension d_2 exists for the correlation integral method; or $\lambda_1 > 0$ for the Lyapunov exponent method; or $K_2 > 0$ for the Kolmogorov entropy method. However, it is revealed that the algorithm of Wolf et al. (1985) may sometimes give positive Lyapunov exponents for random and ARMA processes. Similar observations have also been made for K_2 entropy. Therefore, the correlation integral approach is the most reliable, where D_2 can be identified by computation of d_2 if the studied series is chaotic. The d_2 can be computed by the correlation integral method (depicted in Chapter 3) where delay vectors are first required. The generation of delay vectors relies on τ and m . τ , also called decorrelation time, has to be first determined when m is treated as a variable (generally varying from 2 to 20). Generally, τ can be defined when the auto-correlation function (ACF) attains the value of zero or below a small value, or the Average Mutual Information (AMI) reaches the first minimum (Tsonis, 1992; Abarbanel, 1996; Kantz and Schreiber, 2004). Computations of ACF and AMI have been in detail discussed in the works of Fraser and Swinney (1986), Tsonis (1992), and Abarbanel et al. (1993). If the plot of d_2 versus m tends to level off with the increase of m , d_2 reaches a saturation value D_2 . The real m can be identified at the point where d_2 first reaches D_2 . In the meantime, a direct method to identify m is by false nearest neighbors (FNN) (Wang et al., 2006b). Some researchers also proposed

that τ and m should be concurrently identified (Babovic et al., 2000; Liong and Sivapragasam, 2002; Phoon et al., 2002). For example, Liong and Sivapragasam (2002) used genetic algorithm (GA) to optimize the triplet (m, τ, K).

Depending on delay vectors, the $f(\cdot)$ in Eq. (2.4) can be estimated by two ways, namely, global and local. According to the global approach, a unique predictor is estimated which is valid for any reconstructed vector and depends on the observations at all points. ANNs are usually applied as global models. In contrast, the local approach estimates the $f(\cdot)$ depending on observations that are in some finite neighborhood to the point of estimate. A typical example for local models is nonlinear (NL) prediction originally developed by Farmer and Sidorowich (1987). As a matter of fact, the local prediction approach for a chaotic time series is completely equivalent to the K-NN method. In the local method, the central task is to find the optimal K . Sugihara and Mary (1990) suggested $K = m + 1$. Jayawardena and Lai (1994) ascertained K by trial and error, where five local prediction approaches were examined according to K being equal to 1 or larger than 1: zeroth order approximation when $K = 1$; for $K > 1$, the other four are respectively average K neighbors, weighted average of K neighbors, exponential weighting K neighbors, and linear approximation via fitting K pairs $(Y(t_i), Y(t_i + \tau))$. In contrast, as mentioned previously, some authors recommended that determination of K should be jointly made with the selection of τ and m , which yields the least prediction error (Babovic et al., 2000; Liong and Sivapragasam, 2002).

2.2.3 Applications

Applications of the chaos theory in hydrology have been documented (Rodriguez-Iturbe et al., 1989; Jayawardena and Lai, 1994; Porporato and Ridolfi, 1997; Sivakumar et al., 1998; Babovic et al., 2000; Solomatine et al., 2000; Elshorbagy et al., 2002; Sivakumar et al., 2002). Some studies have investigated the existence of chaos in rainfall, runoff and water level time series whereas the other studies have used the chaos theory to carry out various predictions. Rodriguez-Iturbe et al. (1989) analyzed rainfall data, measured at 15-second time intervals, of a

particular storm event, and Sivakumar et al. (1998) investigated daily rainfall for the existence of chaotic behavior. Jayawardena and Lai (1994) investigated daily rainfall and streamflow time series and used the chaos reconstruction method for rainfall and streamflow predictions in comparison to ARMA. Porporato and Ridolfi (1997) used the chaos theory to predict river flow. Sivakumar et al. (2002) compared the chaos-theory based method (called phase-space-reconstruction (PSR) method) with ANN in the context of daily river flow prediction. Results showed that the PSR method performed better than ANN at all prediction horizons. Elshorbagy et al. (2002) used the principle of temporal evolution of chaotic data to estimate the missing streamflow data. Additionally, the chaos theory-based approach was also applied for water level predictions by Babovic et al (2000) and Solomatine et al. (2000).

2.2.4 Advantages and disadvantages

Regarding the existence of chaos in hydrologic data, some doubts have been raised (Ghilardi and Rosso, 1990; Koutsoyiannis and Pachakis, 1996; Pasternack, 1999; Schertzer et al., 2002; Wang et al., 2006a). Therefore, the chaos theory-based method has no universal applicability. It can be successfully applied only when the time series under study have certain properties, for example, the time series is periodic or indeed exhibits properties of chaotic behavior (or close to it) and it is of adequate length (Solomatine and Ostfeld, 2008).

2.3 Artificial neural networks

2.3.1 Introduction

ANNs mimic the functioning of the human brain and nervous systems in a simplified computational form. They are constituted by highly interconnected simple elements (termed artificial neurons or sometimes termed nodes) which receive information, elaborate them through mathematical functions and pass them to other artificial neurons (Alvisi et al., 2006). They acquire knowledge through a learning process that involves finding an optimal set of weights for the connections and threshold values

for the neurons. ANNs can be categorized as single layer, bilayer and multilayer according to the number of layers, and as feed-forward, recurrent, and self-organizing according to the direction of information flow and processing (ASCE, 2000a). The most widely-used ANNs are the multilayer feed-forward networks, representative examples of which are the multilayer perceptron (MLP) and the radial basis functions (RBF). In particular, the MLP network frequently appears in hydrologic forecasting. Therefore, this section pays primary attention to the MLP network. Other popular ANNs applied in hydrology include RBF (e.g., ASCE, 2000a; Lin and Chen, 2004), Bayesian neural network (BNN) (e.g., Sarle, 1995; Lampinen and Vehtari, 2001; Khau and Coulibaly, 2006), and Stacked neural network (SNN) (e.g., Wolpert, 1992; Breiman, 1994; Shao, et al., 1997).

A MLP network consists of an input layer, one or several hidden layers, and an output layer. Figure 2.1 presents a three-layer feed-forward neural network with each layer having one or more neurons (or nodes). The nodes in one layer are connected to those in the next, but not to those in the same layer. These solid lines describe how the information passes from the input to the output side. The node is the hub of information processing, and its structure is displayed in Figure 2.2. Each node comprises a number of input arcs ($u_1, u_2, \dots, u_n, 1$) and the output arcs. A neuron computes an output, based on a weighted sum of all its inputs (net_j), according to an activation function ($f(net_j)$). Depending on the type of network, training algorithm and the way of data scaling, these activation functions may be logistic sigmoid, linear, threshold, Gaussian or hyperbolic tangent sigmoid functions. In the majority of hydrologic studies, logistic sigmoid function and hyperbolic tangent sigmoid function are used. The connection weight (w_j) reflects contribution of each input arc to the neuron, and the parameter θ_j is a threshold value (also termed bias) which must be exceeded before the node can be activated. These parameters can be determined by network training with a number of training samples.

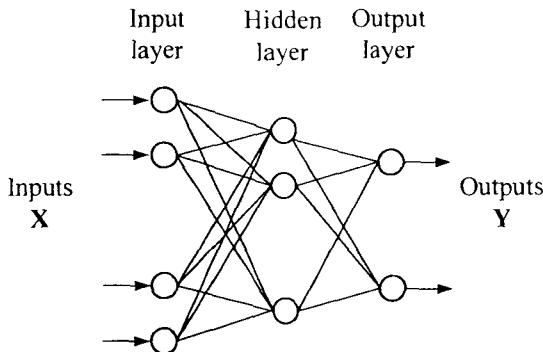


Figure 2.1 A feed-forward ANN structure

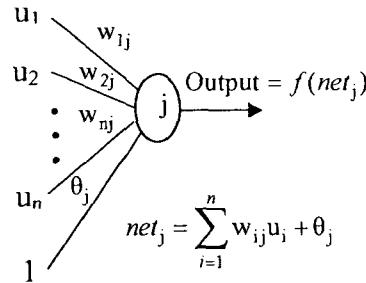


Figure 2.2 Activation of a single neuron

2.3.2 Issues related to model identification

The development of the MLP hydrological forecasting model (hereafter referred to as ANN) can follow the steps recommended by Maier and Dandy (2000) or the procedure designed by Dawson and Wilby (2001). An important difference between them is the sequence between data division and data preprocessing. Burden et al. (1997) stressed that the available data need to be divided into subsets (e.g. training, testing and validation) before any data pre-processing is carried out, which can keep each dataset independent. According to Maier and Dandy (2000) and Dawson and Wilby (2001), model development mainly comprises data gathering, data division , data preprocessing (e.g., data cleaning and data standardization/normalization), determination of model inputs, selection of ANN (e.g., model type, transfer function and training algorithm), network training (e.g., architecture design and training), and evaluation. It should be noted that these processes are not strictly sequential and exclusive, but oftentimes mixed.

To begin with, most hydrologic data are obtained either from gauges that are placed on site or by remote sensing instrument (ASCE, 2000a). The accuracy and robustness of a neural network model is largely influenced by the amount of available training data. An optimal data set should be of adequately large size and representative of the probable occurrence of an input vector and should facilitate the mapping of the underlying nonlinear process. Inclusion of unnecessary patterns could slow network learning. In contrast, an insufficient data set could lead to poor learning or overfitting.

Carpenter and Barthelemy (1994) advocated that the number of input-output data pairs used for training should be equal to or greater than the number of parameters (weights) in the network.

It is common practice to split the available data into two subsets: a training set and an independent validation set. It is also imperative that the training and validation sets are representative of the same population because ANNs are unable to extrapolate beyond the range of the data used for training (Minns and Hall, 1996). Generally, poor forecasts/predictions can be expected if the validation data contains values outside the range of those used for training. Often statistic properties (mean, variance, range) from them are compared in order to measure the representative. When limited data are available, it might be difficult to form a representative validation set. One method which maximizes utilization of the available data is the holdout method (Masters, 1993). The basic idea is to withhold a small subset of the data for validation and to train the network on the remaining data. Once the generalization ability of the trained network has been obtained with the help of the validation set, a different subset of the data is withheld and the above process is repeated. Different subsets are withheld in turn, until the generalization ability has been determined for all of the available data. Generally, the small subset is derived from S equal-sized splitting of the available data. Typical values for S are 5 and 10 (Schalkoff, 1997). Other methods for maximizing the available data for training have been reviewed by Maier and Dandy (2000). On the other hand, from a viewpoint of avoiding overfitting, the available data need to be divided into three subsets: a training data, a testing set (cross-validation) and a validation set. The training set is used to train a number of different ANN model configurations. The test set is used to decide when to stop training (to avoid overfitting) and also to determine which of the networks is the most accurate. Finally, the validation set is used to evaluate the chosen model against independent data.

Data preprocessing herein includes data cleaning and data standardization/normalization. Data cleaning generally consists of data transformation, detrending, and removing seasonal components, and filtering, which attempts to induce a simple systematic relationship between predictors and predictand(s). Whether data

transformation is entailed remains an argument. Some researchers considered that ANN models are not significantly different from a number of statistical models (Hill et al., 1994; Chon and Cohen, 1997). Therefore, the rules governing the traditional statistical models, in particular need of the data being normally distributed, should be considered in ANN modeling (Sudeer, et al., 2002). If the need cannot be met, suitable transformations to normality have to be found. Conventionally-used normalization techniques include log normal, Pearson type III, Wilson-Hilferty and square root transformations. Raman and Sunilkumar (1995) and Sudheer et al. (2002) adapted the Wilson-Hilferty transformation to normalize the flow series. On the other hand, Burke and Ignizio (1992) suggested that ANNs can overcome this problem. Faraway and Chatfield (1998) also pointed out that the model fits were the same irrespective of whether raw or transformed data were used. Obviously, this issue requires further investigation. Conversely, there are seemingly more consistent in support of the requirement of stationarity of data used in ANN. It is generally accepted that ANNs cannot extrapolate beyond the range of the training data. Consequently, it is unlikely that ANNs can account for trends and heteroscedasticity in the data. Until recently, the techniques for dealing with non-stationary sources of data are not as highly developed, nor as well established as those for static forecasting problems (Cannas et al., 2006). One way to deal with this problem is to remove any deterministic components using methods commonly used in time series modeling such as differencing, classical decomposition, and adaptive weight update strategy (Maier and Dandy, 2000). For example, Chng et al. (1996) used the differencing method for non-stationary time series modeling. Cannas et al. (2006) employed wavelet transforms and data partitioning as data preprocessing techniques to improve the performance of ANN in the context of predicting non stationary monthly flows. Regarding seasonal components, removing them seems unnecessary. The study from Hansen and Nelson (1997) has shown that the ability of ANNs to find non-linear patterns in data renders them well suited to dealing with time series with non-regular cyclic variation. Maier and Dandy (1997) investigated the effect of input data with and without seasonal variation on the performance of ANN models. Their results indicated that ANNs have the ability to cater for irregular seasonal variation in the data with the aid of their hidden layer nodes. To some extent, hydrologic time series can be viewed as quasi-periodic signal which is contaminated

by various noises at different flow levels. If suitable noise-reduction techniques can remove most of noises, the processed data set should have better predictability than the raw data set (Lisi et al., 1995). As such techniques, wavelet transforms (WT) and singular spectrum analysis (SSA) have been reported in the field of water resources (Sivapragasam et al., 2001; Baratta, et al., 2003; Küçük and Ağaloğlu, 2006; Partal and Kişi, 2007; Partal and Cigizoglu, 2008; Adarnowski, 2008; Partal, 2009). For example, Sivapragasam et al. (2001) established a hybrid model of support vector machine (SVM) in conjunction with the SSA for predictions of rainfall and runoff. Baratta et al. (2003) developed an ensemble ANN model for daily rainfall forecasting by grouping reconstructed components (RCs) of SSA. Applications of discrete wavelet transforms (DWT) to precipitation and discharge were presented by Partal and Kişi (2007) and Partal and Cigizoglu (2008) respectively, where DWT was applied to each model input variable. Results indicated that DWT is highly promising for improvement of the model's performance. To date, applications of these noise-reduction techniques, in particular SSA, to the field of hydrology, are not common, and performance of these techniques is also worthwhile to be compared.

All variables should be standardized to ensure them receive equal attention during the training process (Maier and Dandy, 2000). Otherwise, input variables measured on different scales will dominate training to a greater or lesser extent because initial weights within a network are randomized to the same finite range (Dawson and Wilby, 2001). In addition, the variables have to be scaled in such a way as to be commensurate with the limits of the activation functions used in the output layer (Minns and Hall, 1996). In general, data are rescaled to the intervals [-1, 1], [0, 1] or [0.1, 0.9] or others (referred to as standardization) depending on the type of activation functions (Rajurkar et al., 2002; Cannas et al., 2002; Campolo et al., 2003). Another approach is to rescale values to a Gaussian function with a mean of 0 and unit standard deviation (referred to as normalization), which is by subtracting the mean and dividing by the standard deviation. The advantage of using [0.1, 0.9] for runoff modeling is that extreme (high and low) flow events, occurring outside the range of the calibration data, may be accommodated (Hsu et al., 1995). Moreover, if the values are scaled to the extreme limits of the transfer function, the size of the weight updates is extremely small and flatspots in training are likely to occur (Maier

and Dandy, 2000). It is worth noting that scaling is not strictly required when the transfer functions in the output layer are unbounded. However, scaling to uniform ranges is still recommended (Masters, 1993). To investigate the influence of data preprocessing on model performance in more depth, Wang et al. (2006b) compared six data preprocessing schemes (viz. standardization, normalization, standardization of log-transformation, normalization of log-transformation, deseasonalization of log-transformation, and normalization to deseasonalization of log-transformation) using hybrid neural networks for flow time series predictions. The scheme of deseasonalization of log-transformation was found the most effective. Similar study should be extended to multivariate time series with different seasons (hourly, daily or monthly data).

Selection of appropriate model inputs is extremely important in the model building process. A wide range of methods of input determination have been used by the hydrological community and were reviewed by Bowden et al. (2005). These methods are either model free or model dependent. The model dependent methods are to use a model to characterize the relationship between input and output whereas the model free methods do not depend in any way on a pre-existing model. Six methods can be categorized on a literature review basis. They are derived from (a) *a priori* knowledge of the modeled system; (b) principal component analysis(PCA); (c) correlation analysis; (d) a heuristic approach (or trial and error); (e) knowledge extraction; (f) composite methods.

The majority of research papers for rainfall-runoff transformation rely on the use of *a priori* knowledge to select the appropriate model inputs (Minns and Hall, 1996; Thirumalaiah and Deo, 2000). Generally, some degree of *a priori* knowledge of causal variables is necessary to define a candidate set of inputs. However, it is dependent on an expert's knowledge, and hence is very subjective and case-dependent.

When the candidate set of inputs is large, PCA is often employed (Roadknight et al., 1997; Guhathakurta et al., 1999; Hu et al., 2007; Al-Alawi et al., 2008). It can obtain a small set of linear independent inputs which still retain most of the total variance of

original inputs. For example, Hu et al. (2007) employed PCA as an input data-preprocessing tool for ANN to improve the accuracy of the rainfall-runoff prediction. However, PCA was only effective for spatially distributed gauges as rainfall data instead of a basin-averaged value.

When the relationship to be modeled is less well understood, various analytical techniques can be used, in particular for univariate time series. Some researchers used linear correlation analysis such as cross correlation (CC), autocorrelation (AC), and partial autocorrelation (PAC) for input selection (Yu et al., 2000; Luk et al., 2000; Sebzalli and Wan, 2001; Sudheer et al., 2002; Abrahart, 2003; Anctil and Lauzon, 2004; Jain et al., 2004; de Vos and Rientjes, 2005; Kumar et al., 2005; Nayak et al., 2005; Chetan and Sudheer, 2006; Abrahart and See, 2007; Nayak et al., 2007; Mutlu et al., 2008; Solomatine et al., 2008). Other researchers advocated the use of nonlinear correlation analysis including average mutual information (AMI), and partial mutual information (PMI) because linear correlation analysis is unable to capture any nonlinear dependence which can widely exist in rainfall-runoff transformation (Bowden et al., 2005; May et al., 2008; Solomatine et al., 2008).

As model dependent approaches, the heuristic approach (or trial-and-error) is also commonly used in the literature. In this approach, various ANN models are trained using different subsets of inputs (Hsu, et al. 1995; Raman and Sunikumar, 1995; Fernando and Jayawardena, 1998; Thirumalaiah and Deo, 1998; Jain et al., 1999; Tokar and Johnson, 1999; Cannas et al., 2002; Wilby et al., 2003; Chattopadhyay and Chattopadhyay, 2007). To avoid enumerating all input combinations, a stepwise approach is used in which separate networks are trained for each input variable (Masters, 1993; Maier et al., 1998). Main drawbacks of the heuristic method are that there is no guarantee that they will find the globally best subsets, and are computationally intensive.

Another type of model dependent method is knowledge extraction from the trained ANN. Sensitivity analysis is the most commonly-used method of knowledge extraction (Maier and Dandy, 1997; Nord and Jacobsson, 1998; Liou et al., 2000; Maier and Dandy, 2000; Gevrey et al., 2003; Sudheer, 2004; Gevrey et al., 2006,

Papadokonstantakis et al., 2006; Mutti and Chau, 2006). Plots of the sensitivities for each of the inputs are usually inspected and the significant inputs are chosen using judgment. However, the difficulty with this approach is choosing a reasonable value to perturb the input by selecting the appropriate cut-off point for input significance.

In practice, these methods are jointly used and referred to as the composite method. For instance, PCA is combined with *a priori* knowledge or analytical techniques. Silverman and Dracup (2000) used *a priori* knowledge of the system under investigation in conjunction with the trial-and-error approach to determine inputs for long-range forecasts of precipitation in California using ANNs. Therefore, the recommended approach for determining appropriate inputs and lags of inputs should be a combination of *a priori* knowledge and other methods mentioned above (see Sajikumar and Thandaveswara, 1999; Campolo et al., 2003; Solomatine and Dulal, 2003; Yu et al, 2006).

In addition, some global optimization methods are also used to automatically select optimal inputs. For example, Abrahart et al. (1999) used a genetic algorithm (GA) to optimize the inputs to an ANN model used to forecast runoff from a small catchment. Giustolisi and Simeone (2006) employed a multi-objective genetic algorithm to simultaneously optimize model inputs and the ANN's architecture. However, there are few studies performing a comparison of these methods of inputs selection. For example, Sudheer et al. (2002) compared the linear correlation analysis method with the heuristic method. It is recommended that a comparison be imperative in model inputs determination because model inputs considerably influence the performance of ANN.

The architecture design of ANN consists of the number of hidden layers and the number of neurons in input layer, hidden layers and output layer. ANN with one hidden layer are commonly used in hydrologic modeling since these networks are considered to provide enough complexity to accurately simulate the dynamic and nonlinear-properties of the hydrologic process (Hornik et al., 1989; Dawson and Wilby, 2001; de Vos and Rientjes, 2005). Therefore, the architecture design consists in either specifying model inputs and the number of nodes in the hidden layer, or

only determining the number of nodes in the hidden layer if model inputs are already determined. There are some algorithms, including regularization, pruning, constructive, and stopped training (Anders and Korn, 1999), to determine optimal network geometries. However, a trial and error procedure using different number of neurons is still the preferred choice of most users (e.g. Shamseldin, 1997; Zealand et al., 1999; Abrahart and See, 2000). A main disadvantage of regularization, pruning and stopped training is that these methods contain a strong judgmental component, which makes the model building process difficult to reconstruct (*ibid*). Anders and Korn (1999) developed model selection strategies on the basis of statistical concepts, i.e., hypothesis testing, information criteria and cross validation methods. Chetan et al. (2006) suggested online techniques, both constructive algorithms and pruning algorithms, do not ensure the reaching of the best topology. In two online techniques, decision on whether the addition or removal of a node has an impact on the performance of the network is generally assessed using the training data, which does not give an accurate indication of the generalization ability of the network (Doering et al. 1997). However, this problem can be overcome by using some of the measures of generalization ability such as early-stopping. Another disadvantage of pruning and constructive approaches is that networks generally have to be trained a number of times (Maier and Dandy, 2000). Recently, GAs have successfully been used to determine optimal network architectures (Abrahart et al., 1999; Zhang, 1999; Giustolisi and Simeone, 2006).

The ANN training is a problem of nonlinear optimization, which searches the optimal ANN's connection weights by minimizing the error between the network output and the target output. Training algorithms are local or global. The best-known training algorithm in local algorithms is the classical back-propagation algorithm (Rumelhart *et al.*, 1986), which follows a steepest-descent approach based on the first-order gradient of the slope of the objective function. The algorithm is susceptible to local minima. Other popular back-propagation methods include the conjugate gradient algorithm and the Levenberg-Marquardt (L-M) algorithm. The L-M algorithm belongs to the quasi-Newton method which is the modification of classical Newton algorithm by using an approximation of the inverse of the Hessian (Fletcher, 1987). The approximation can guarantee that the Hessian is positive definite while

maintaining a convergence rate of order two. The second-order convergence renders it possibly to escape local minima in the error surface. As a fast and efficient algorithm, the L-M algorithm is therefore widely adopted by researchers (Anctil et al, 2004; Huang et al., 2004; Kisi, 2004; Smaoui and Al-Enezi, 2004; Coulibaly et al., 2005; de Vos and Rientjes, 2005; Giustolisi and Laucelli, 2005; Jeong and Kim 2005; Aqil et al., 2007; Pulido-Calvo and Portela, 2007). In the meantime, global optimization techniques such as the Particle Swarm Optimization (PSO), the Genetic algorithm (GA), and the Shuffled Complex Evolution (SCE-UA), are becoming popular (Chau, 2006; Chau et al., 2005; Lin et al, 2006). In comparison with the gradient-based techniques, these algorithms are able to overcome local minima and obtain more stable solutions, but require more computational effort. As an attempt to combine their virtues of the two optimization techniques, an integrated training algorithm combining L-M with GA was adopted in Chau et al. (2005) where GA was used to search an initial set of weights and biases.

In the training of, a very important issue is how to decide when to stop the training because ANNs are prone to either underfitting or overfitting if their trainings are not stopped appropriately. The cross-validation early stopping technique is most commonly used (see, ASCE, 2000a) to avoid overfitting. However, some researchers (e.g. Masters, 1993; Ripley, 1994) suggested that it is impossible to determine the optimal stopping time and there is a danger that training is stopped prematurely. Amari et al. (1997) showed that overfitting would not occur if the ratio of the number of training set to the number of the weights in the network exceeds 30. In such cases, training can be stopped when the training error has reached a sufficiently small value or when changes in the training error remain small. Wang et al. (2005) found that very slight overfitting still can be observed in some cases when the ratio is as high as 50. In summary, how to stop training still remains quite subjective. With respect to model validation, it should be noticed that the validation data should not have been used as part of the training process (Maier and Dandy, 2000).

2.3.3 Applications

Due to a highly flexible structure, ANNs are able to capture any complex mapping

relationship between input and output variables, which has led to the applications of ANNs in a wide spectrum of hydrological forecasting issues.

ANNs have been embraced enthusiastically by hydrologic practitioners since the early nineties (Maier and Dandy, 2000). They have been successfully used in hydrology-related areas such as rainfall estimation, rainfall-runoff modeling, streamflow forecasting, ground-water modeling, water quality, water management policy, precipitation forecasting, hydrologic time series, reservoir operations and uncertainty prediction (ASCE, 2000b; Maier and Dandy, 2000; Dawson and Wilby, 2001).

ANN is often used for quantitative rainfall prediction instead of complex mechanistic models. Monthly and seasonal all India summer monsoon rainfall were predicted using ANN modeling by Sahai et al. (2000). Brath et al. (2002) employed ANN and K-NN to predict rainfall which is used as the input of conceptual model to improve flood forecasting. Cannon and McKendry (2002) compared ANN and multiple linear regression (MLR) models in Indian monsoon rainfall prediction. Guhathakurta (2008) employed ANN to carry out multi-step monthly rainfall prediction. Similar studies include Silverman and Dracup (2000) and Chattopadhyay and Chattopadhyay (2007).

While some researchers employed ANNs solely for river flow forecasts (Prochazka, 1997; Thirwmalaiah and Deo, 1998; Sheta and El-Sherif, 1999; Liong et al., 2000; Imrie et al., 2000; Salas et al., 2000; Deo and Thirwmalaiah, 2000; Qin et al., 2002; Cannon and Whitfield, 2002; Li and Gu, 2003; Sudheer and Jain, 2003; Huang et al., 2004; García-Pedrajas et al., 2006), other researchers compared ANNs with traditional statistical techniques for flood predictions in rivers. Comparisons between ANNs and AR appeared in the work of Raman and Sunilkumar (1995), Thirumalaiah and Deo (2000) and Kişi (2003, 2005). Likewise, some studies focused on comparisons between ANNs and ARMA (Jain et al., 1999; Abrahart and See, 2002; Castellano-Méndez et al., 2004). The majority of studies have proven that ANNs outperform traditional statistical techniques. Furthermore, the superiority of ANNs over linear regression in predicting river flows attributes to the possible existence of

nonlinear dynamics in studied series, which are not well captured by the regression technique. In addition, some authors employed modular or hybrid techniques to simulate streamflow. For instance, Abrahart and See (2000) presented a hybrid network solution based the clustering of the hydrological records with a self-organizing map (SOM) neural network. Hu et al. (2001) developed range-dependent hybrid neural networks (RDNN), which are virtually threshold ANNs, to forecast annual and daily streamflows. Chang et al. (2002) developed a fuzzy clustering neural networks model for flood forecasting. Islam et al. (2001) and Sivakumar et al. (2002) used ANNs in combination with the PSR technique to predict river flow. A hybrid ANN model developed by Wang et al. (2006b) was used to predict daily streamflow. Corzo and Solomatine (2007) addressed the baseflow separation using a modular ANN.

ANNs are also used for modeling the rainfall/runoff relationship. The most popular is conventional error back propagation neural network (Ichijanagi et al., 1995; Hsu et al., 1995; Smith and Eli, 1995; Tokar and Markus, 2000; Sudheer et al., 2002; Rajurkar et al., 2002; Campodo et al., 2003; Seno et al., 2003; Anctil et al., 2004; Wu et al., 2005; de Vos and Rientjes, 2005). Chang and Chang (2001) and Lin and Chen (2004) employed RBF neural networks to address flood forecasting. Chang et al. (2002) developed a recurrent ANN (also termed dynamic ANN) for real-time streamflow predictions. Khan and Coulibaly (2006) employed Bayesian neural network to estimate the uncertainty of rainfall/runoff relationship. However, de Vos and Rientjes (2005) emphasized the issue of lagged predictions in ANN model. They recommended that the issue may be mitigated by moving average over inputs or multiple measures of model performance during ANN training. There are also some examples of applications of modular or hybrid ANN models in rainfall-runoff transformation. Zhang and Govindaraju (2000) examined the performance of modular networks in predicting monthly discharges based on the Bayesian concept. Besides, neural networks are also coupled with other techniques, such as fuzzy set, genetic algorithm, and genetic programming, for rainfall/runoff predictions (Chang and Chen, 2001; Rabuñal et al., 2003; Jain and Srinivasulu, 2004; Chetan and Sudheer, 2006).

Few quantities of papers are concerned with reservoir (Raman and Chandramouli 1996; Chibanga et al., 2003; Baratti et al., 2003). For example, Raman and Chandramouli (1996) used ANN to derive reservoir operation policy. Chibanga et al. (2003) employed ANNs to forecast the inflow of reservoir.

ANNs are also used as assistant tools in hydrological forecasting. ANNs as error updating components for conceptual rainfall/runoff models or data-driven models have been documented in the literature (Babovic et al., 2001; Brath et al., 2002; Vojinovic et al., 2002; Anctil et al., 2003; Al-Alawi et al., 2008). Shamseldin et al. (2007) compared three types of ANNs which are used as three different combination methods to address ensemble rainfall-runoff prediction. Al-Alawi et al. (2008) used ANN to conduct an error updating of principal component regression model.

Recently, ANNs are involved in uncertainty of prediction (or interval prediction). For my best knowledge, the first study may be conducted by Chryssoloiuris et al. (1996) who use several methods to determine the confidence interval of prediction of ANN. ANN is used as a surrogate model of numerical models or conceptual hydrological models to reduce Monte Carlo simulation in uncertainty estimation (Zou et al., 2002; Khu and Werner, 2003; Shrestha et al, 2009). Shrestha and Solomatine (2006) and Solomatine and Shrestha (2009) created a novel method to estimate prediction interval of the model output. The method is to use ANN to map the relationship between hydro-meteorological variables and prediction errors.

Along with the question that “Can ANNs be made to reveal any physics?” offered by ASCE (2000b), many researchers have carried out studies to provide insight into unrecognized relationships within hydrological “black boxes”. There are two directions to construct grey neural network models. One is based on the influence analysis of input variables (Sudheer, 2004). Corresponding methods were reviewed and compared by Papadokonstantakis et al. (2006). The other is to attempt to match physical significance with the internal architecture of ANN hydrologic models (Wilby et al., 2003; Jain and Srinivasulu, 2004; Sudheer and Jain, 2004, Ahmad and Simonovic, 2005; Jain and Srinivasulu, 2006). Sudheer and Jain (2004) suggested that three hidden neurons in the ANN river flow model correspondingly map the

higher, lower, and middle magnitude portions of the transfer function respectively although some portions are overlapping. Ahmad and Simonovic (2005) used ANN to simulate a hydrograph.

2.3.4 Advantages and disadvantages

The greatest advantage of ANN over traditional modelling techniques is a capability to model complex, non-linear processes without having to assume the form of the relationship between input and output variables. This, in conjunction with the fact that ANNs are able to self-adjust to information, requires little expertise of the problem under consideration for their successful applications. Moreover, due to their compact and flexible model structure, ANNs have relatively low computational demands and can easily be integrated with other techniques.

However, ANN still has a salient drawback of not being transparent, in the sense that its functioning is rather obscure and not easily interpretable (Jacquin and Shamseldin, 2009). This is a significant weakness, for without the ability to produce comprehensible decisions it is hard to build up the confidence or reliability of networks in addressing real-world problems (Benitez et al. 1997). Another important drawback is that the optimal form or value of most network design parameters (such as the number of neurons in the hidden layer) can differ for each application and cannot be theoretically defined, which is why they are commonly determined using trial-and-error approaches (de Vos and Rientjes, 2005). Furthermore, parameter estimation and over-fitting problems represent the principal disadvantages of model construction by ANN, as reported in Giustolisi and Laucelli (2005). In addition, ANN does not allow knowledge derived from known physical laws to be incorporated into the learning process.

2.4 Fuzzy inference systems

2.4.1 Introduction

The fuzzy inference systems (FIS), originated from the pioneer contributions of Zadeh (1965, 1994), are nonlinear black-box models that describe the relation between the inputs and the output of a real system using a set of fuzzy IF-THEN rules. The basic concepts that comprise FIS, such as fuzzy set, membership functions, the domain partitions, and fuzzy if-then inference rules, are not detailed herein as they have been presented in numerous hydrological papers and text books (Tanaka, 1996; Abe, 1997; Clair et al., 1997; Novák et al., 1999; Cordón et al., 2001; Kecman, 2001; Xiong et al., 2001; Maskey et al., 2004). The structure of a fuzzy rule can be formally expressed as

If (X_1 is $A_{1,m}$) AND (X_2 is $A_{2,m}$) AND ... AND (X_k is $A_{k,m}$) THEN Y is

where $\mathbf{X} = (X_1, X_2, \dots, X_k)$ is the input vector and Y is the output variable of the model. The rule antecedents (viz. the IF part of the rules) are defined by the terms $A_{k,m}$ representing fuzzy sets used to partition the input space into a number of overlapping regions, such that the feasible range of each input variable X_i ($i=1, \dots, k$) is divided into a number of a series of fuzzily defined and overlapping intervals $A_{i,m}$. (The process is termed fuzzification). Each fuzzy set $A_{i,m}$ is described by its membership function $\mu_{i,m}$, which assigns a membership value $\mu_{i,m}(x_i)$ to every possible input $X_i = x_i$. The membership values $\mu_{i,m}(x_i)$ vary in the range [0, 1], where 0 indicates absolute non-membership and 1 indicates full membership of $X_i = x_i$ in the fuzzy set $A_{i,m}$. The structure of the rule consequents (viz. the THEN part of the rule) could be either a fuzzy set (then such a model is Mamdani model (Mamdani, 1974)) or a function, often linear (and then the model is referred to as a Takagi-Sugeno-Kang (TSK) model (Takagi and Sugeno, 1985; Sugeno and Kang, 1988)).

The rule consequences of a Mamdani-type FIS can be formally represented by $Y=B_m$. B_m represents a fuzzy set in the output space. The m th rule expresses a tendency of input vectors $x=(x_1, x_2, \dots, x_k)$ in the region described by $A_{i,m}$ ($i=1, \dots, k$) to be associated with outputs y in the region defined by B_m . For a given input vector

$\mathbf{X} = \mathbf{x} = (x_1, x_2, \dots, x_k)$, the inference mechanisms of fuzzy logic are used to obtain the fuzzy output of each rule and to derive the overall (crisp) output $Y = y$ of the FIS. In the inference process, the AND operator (or termed implication) mathematically applied as an intersection operator by either the “minimum” or “product” function. Minimum is commonly used when the input data are independent of each other, and product is often applied if input variables are interdependent (Robison Fayek and Sun, 2001). The results of all of the rules can be combined in a process termed aggregation. “Sum” and “maximum” are two conventional operator methods for aggregation. If the sets from implication are summed together, then the method of aggregation is termed summation. If aggregation of the sets occurs by combining the maximum values obtained for each output membership function after implication, then maximum method has been used. The final process is defuzzification where the overall fuzzy output is converted into a single crisp value. Conventional defuzzification methods involve the centre of area (or gravity) method, bisector method, and other some methods focusing on the maximum membership value attained by the solution set.

In the case of TSK-type FIS, the rule consequents are explicit functions of the input variables. The consequent of the m th rule is expressed as $Y = f_m(X_1, X_2, \dots, X_k)$. The functions f_m are given as $f_m(X_1, X_2, \dots, X_k) = b_{0,m} + b_{1,m}X_1 + b_{2,m}X_2 + \dots + b_{k,m}X_k$, which demonstrate that the local models are linear functions of the input variables to the system. The non-linearity of the overall fuzzy model response comes from the interaction between the rules. Since the conclusion of each IF-THEN inference rule is a scalar rather than a fuzzy set for the control variable, defuzzification procedures are not required. Therefore, the inference process in TSK method can be simply summarized into three steps: fuzzification, inference, and aggregation (composition). The overall operation of a multiple-input single-output TSK-type FIS is shown in Figure 2.3. For a given input vector $\mathbf{X} = \mathbf{x} = (x_1, x_2, \dots, x_k)$, the degree of fulfillment (DOF) of each rule is first calculated (also termed implication). The DOF is normally evaluated using a T-norm, such as the algebraic product:

$$DOF_m(x_1, x_2, \dots, x_k) = \mu_{A_{1,m}}(x_1) \cdot \mu_{A_{2,m}}(x_2) \cdot \dots \cdot \mu_{A_{k,m}}(x_k) \quad (2.5)$$

Several types of the membership functions, including triangular functions and Gaussian functions, can be used for the fuzzy sets in the antecedent of the rules. The overall response of the fuzzy system is calculated as a weighted combination of the response of the individual rules. The final output of the TKS system with M rules is given by

$$Y = \frac{\sum_{m=1}^M \text{DOF}_m(x_1, x_2, \dots, x_k) \cdot [b_{0,m} + b_{1,m}x_1 + b_{2,m}x_2 + \dots + b_{k,m}x_k]}{\sum_{i=1}^m \text{DOF}_i(x_1, x_2, \dots, x_k)} \quad (2.6)$$

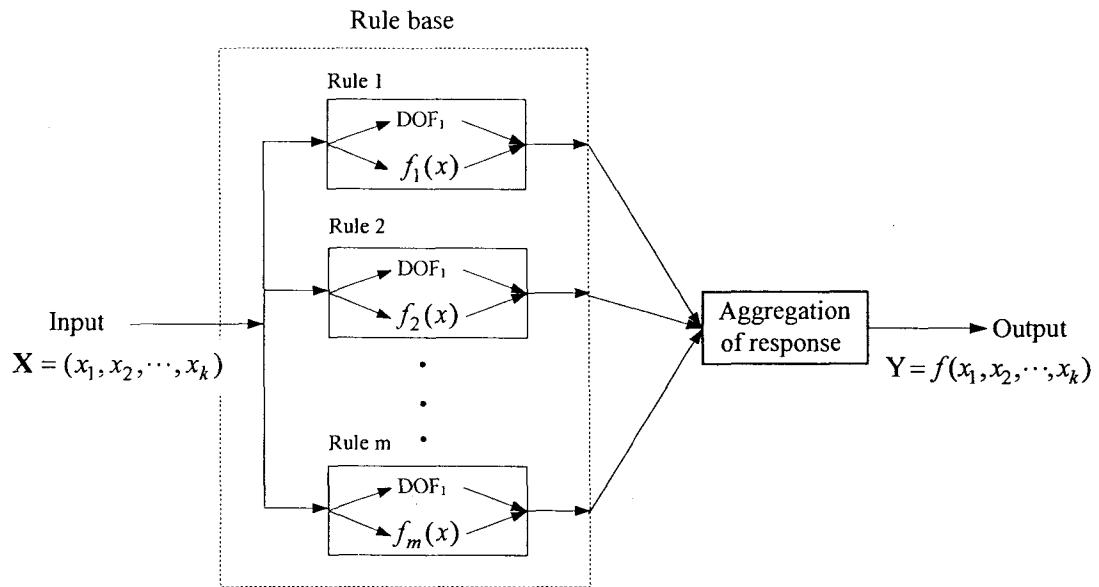


Figure 2.3 TSK fuzzy inference system (adopted from Shu and Burn, 2004)

2.4.2 Issues related to model identification

Identification of the structure of fuzzy logic models is related to many aspects including model inputs, the domain partitions, membership functions, fuzzy rules generation, etc. The strategy to construct hybrid fuzzy modeling has been presented by Valdés et al. (2005).

Similar to ANN, the selection of appropriate input variables is extremely important for FIS since it provides the basic information about the system being modeled. Usually, the number of input variables is known *a priori*. A firm understanding of the hydrologic system under consideration plays an important role in the successful implementation of the model. This helps in avoiding loss of information that may

result if key input variables are omitted, and also prevents inclusion of spurious input variables that tend to confuse the training process.

The rules in FIS increase exponentially with the increase of input variables or membership functions, which tends to cause curse of dimensionality. The addition of unnecessary variables and membership functions would create a more complex model than necessarily required, and may lead to problems defining rules. Moreover, the complex model is susceptible to overfitting of training data. Therefore, variables independent of each other as system inputs are imperative. A number of methods to determine inputs for FIS have been presented in the literature. Statistical analysis is the commonly-used tool (Mahabir et al., 2003; Nayak et al., 2005, Keskin et al., 2005; Aqil et al., 2006). For example, Mahabir et al. (2003) employed the cross correlation analysis to delete redundant precipitation stations. Instead of statistical analysis methods which cannot capture nonlinear dependences, a trial-and-error procedure by composing different inputs is used to select optimal inputs according to the analysis of residuals from different models (Nayak et al., 2004). In addition, Liong et al. (2000) employed two sensitivity analysis approaches to refine inputs.

A FIS can be viewed as a partition in the multidimensional feature space, where the number of partitions in each dimension corresponds to the number of fuzzy sets and the corresponding membership function that are defined in that dimension (Nayak et al., 2005). Consequently, the input space partitioning plays a major role in the optimal architecture of the model. Input space partitioning can be conducted in different ways: grid partitioning and fuzzy clustering. Conventional fuzzy clustering involves subtractive clustering (Chiu, 1994) and Gustafson Kessel (GK) clustering (Gustafson and Kessel, 1979). The major drawbacks of grid partitioning are that the number of rules grows exponentially (Nayak et al., 2005) and that the membership functions for every variable are constructed independently of each other, causing that the relationship between variables is neglected (Vernieuwe et al., 2005). Moreover, the optimization of antecedent parameter becomes complex if grid partitioning is employed. In fuzzy clustering method, the antecedent parameters are obtained directly from the fuzzy clusters. A limitation of this clustering based FIS models is that if any data point falls away from the cluster centre or outside the clusters the

model performance may not be satisfactory. However, grid partitioning method represents the whole range of the input space, and if the complex optimization of antecedent parameters can be properly addressed, the resulting FIS may perform better than clustering based FIS (Nayak et al., 2005). Vernieuwe et al. (2005) compared the three domain partitions using a TSK model, and the results showed that the best results are obtained for the GK clustering method. A worthy noting issue is that data points have to be rescaled to [0, 1] for subtractive clustering method.

There is no fixed guidance to select membership functions. Some studies define the membership functions of all antecedent variables using prior knowledge and experience, which usually appear in grid partitioning method FIS. Some define the membership functions depending on Euclidean distance formula, which appears in clustering method FIS. Despic and Simonovic (2000) gave a review on methods for membership construction. Liong et al. (2000) derived membership functions for input variables on the basis of their frequency distributions. Civanlar and Trussell (1986) developed an approach to construct membership functions based on a known probability density function of elements. Some researchers employed genetic algorithms (GAs) to tune membership functions (Karr, 1991a, b; Thrift, 1991; Chang and Wu, 1995; Homaifar and Mc Connick, 1995).

Inference mechanism in the process of implication, aggregation and defuzzification may vary owing to the use of different fuzzy logic operators. Mahabir et al. (2003) employed a sensitivity analysis on implication “AND” operator (termed T-norms), including minimum method and product method, and aggregation “OR” operator (called S-norms) including summation method and maximum method, to determine optimal fuzzy inference. Despic and Simonovic (2000) developed the original aggregation method named polynomial composition under pseudomeasures and gave a review on four well-known aggregation methods including ordered weighted average operators, gamma-operators, composition under pseudo measure and fuzzy algorithmic approach. Common defuzzifications include center-of-area (or center-of-gravity) and maxima methods.

2.4.3 Applications

FIS has been successfully applied to many fields related to water resources engineering, for instance, rainfall and river flow (or river stage) predictions, decision-making on reservoir operation, rainfall/runoff transformation, and as an assisting tool (e.g., Liong et al., 2000; See and Openshaw, 2000; Xiong et al., 2001; Hasebe and Nagayama, 2002).

FIS has been effectively used for river flow forecasting. For example, an adaptive FIS was used to simulate water demand in a river, and showed the same performance as that in ANN and multiple regression methods (Muster et al., 1994). Streamflow predictions using fuzzy logic technique can be found in the work of Liong et al. (2000), Mahabir et al. (2003), Han et al. (2002), and Chang et al.(2005). See and Abrahart (2001) used four data fusion approaches (mean, median values, two ANNs) to combine predicted results from ANN, FL and ARMA to improve river level forecasting. Moreover, Abrahart and See (2002) discussed six alternative data-fusion methods for river flow forecasting. Recently, adaptive neural fuzzy inference system (ANFIS), as a typical hybrid mode of fuzzy logic and neural networks, have widely proposed for river flow prediction (Deka and Chandramouli, 2003; Nayak et al., 2004; Deka and Chandramouli, 2005; Chang et al., 2005; Chau et al., 2005; Keskin et al., 2005). Their results showed that ANFIS performed better at least similarly when compared to ARMA or/and ANNs.

The application of FIS in the development of rainfall-runoff models has also been investigated. See and Openshaw (2000) used a combination of a hybrid neural network, an autoregressive moving average model and a simple fuzzy rule-based model for rainfall-discharge forecasting. Hundecha et al. (2001) developed fuzzy rule-based routines to independently simulate components of snowmelt, evapo-transpiration, runoff, and basin response for the physically-based rainfall/runoff model-HBV. In view of automatically constructing IF-THEN rules from measured data, the first-order Takagi-Sugeno models were popularly used to mimic rainfall-runoff dynamics (Xiong et al., 2001; Nayak et al., 2004; Vernieuwe et al., 2005; Jacquin and Shamseldin, 2006). Yu and Chen (2005) developed a gray

fuzzy model to carry out rainfall forecasting. Tayfur and Singh (2006) developed ANN, FL, and kinematic wave approximation (KWA) models to simulate rainfall-runoff hydrograph, and results showed that ANN and FL models were satisfactory. Similarly, Luchetta and Manetti (2003) used fuzzy logic model to predict rainfall/runoff transformation, and a comparison with ANN approach showed a better performance of the fuzzy technique. Nayak et al. (2007) developed a rainfall/runoff model by integrating FL and ANN and the hybrid model was comparable with ANFIS. The study suggested that the proposed model could be a feasible alternative to ANFIS for use as an operational tool for rainfall-runoff modeling purposes. Finally, ANFIS models were also used to carry out a rainfall/runoff prediction and a comparison with other methods including ANN, FL and ARMA often showed a better performance than ANFIS model (Nayak et al., 2005, Chen et al., 2006). Chau et al. (2005) compared ANN-GA and ANFIS in the context of river stage and close performance was observed between two models.

Another important application field of FIS is reservoir operation. Kojiri and Ikebuchi (1988) employed fuzzy logic to decide real-time release of reservoir using four antecedents including storage volume, inflow discharge, increasing rate of inflow, and predicted total rainfall. A fuzzy expert system for real-time reservoir operation was developed by Kojiri et al. (1992). Similar researches can be found in the work of Hasebe et al. (1992), Shrestha et al. (1997), Hasebe et al. (1998), and Dubrovin et al. (2002). The FIS in conjunction with GA for reservoir management was found in Chang and Chen (1998) and Chang and Chang (2001). Fuzzy dynamic programming was applied to reservoir operation (Moncada et al., 1997). Karaboga et al. (2004) developed a fuzzy control system to control the spillage gate. Bagis and Karaboga (2004) used a hybrid method of ANN and FL to control spillage gate of reservoir, where the lake level and change in the lake level are antecedents and the gate opening is consequent. Hasebe and Nagayama (2002) compared fuzzy system and neural network-fuzzy system for reservoir operation, and results showed that the fuzzy was an effective operation system for regular operation whereas the other one is effective primarily for flood control.

Additionally, fuzzy logic was also used as effective tools for forecast combination,

uncertainty forecasting, and updating of forecasting. See and Openshaw (1999) and Corani and Guariso (2005) employed a fuzzy logic model to combine several ANNs into a single forecasting system. See and Openshaw (2000) used Mamdani-type FIS for combining the outputs of a series of river level forecasting models, finding that this combination method yielded better results than more traditional model combination strategies, namely an unweighted average and a Bayesian approach. Similarly, Xiong et al. (2001) used TSK-type FIS for combining the outputs of five rainfall-runoff models, comparing the performance of this combination method with that of a simple average, a weighted average and an ANN model. Overall, the FIS-based combination was as accurate as the weighted average and the ANN model. Maskey et al. (2004) developed a fuzzy set approach to treat the precipitation uncertainty in rainfall-runoff modeling. Yu and Chen (2005) proposed a fuzzy rule-based model for the update of the real-time flood forecasting, and results showed that the problem of the phase lag could be reasonably mitigated.

The majority of studies have proved that FIS has good forecasting abilities. However, some studies also showed that the forecasting produced by FIS were not really superior to those of far simpler methods, such as ARMA (See and Openshaw, 2000) or multiple linear regressions (Jacquin and Shamseldin, 2004). These findings only confirm the need to evaluate the quality of any model against real data, as “no such “perfect” model exists that will perform well on all catchments, at all times, and in all circumstances” (O’Connor, 2005).

2.4.4 Advantages and disadvantages

An important advantage of FIS over traditional models (viz. model-driven models) is their ability to infer the behavior of complex systems purely from data, without prior specification of a functional structure. An advantage of FIS over ANN (or similar techniques) is that the functioning of FIS is more transparent. Because the structure of fuzzy rules can be extracted from the knowledge available about the real system, they can provide a formal representation with a more readily physical interpretation. In fact, subjective knowledge provided by experts can be incorporated into the model in a natural manner by translation into fuzzy IF–THEN rules, which can then be

consecutively calibrated to fit the data. In the sense that their internal operation can be relatively easily understood by human, FIS are sometimes considered “gray-box” rather than black-box type models (Lindskog, 1997).

In spite of these merits, the use of FIS in river flow forecasting is not as widespread as ANN. One reason is that a fuzzy-rule-based representation is inadequate to provide an accurate description of the input–output relationship in cases where too many input variables are required and/or the input–output relation is too complex. As mentioned previously, one of the main weaknesses of FIS is the curse of dimensionality, which refers to the fact that the number of fuzzy rules increases exponentially with the number of inputs, eventually resulting in a non-parsimonious model which can be very difficult to calibrate. Reducing the number of rules, however, generally decreases the approximation capabilities of the FIS. Evidently, this problem becomes more serious as the complexity of the input–output relationship increases. Another issue affecting the applicability of FIS is the lack of guidelines for calibrating the model parameters in such a manner that their major advantage, namely the model interpretability, is maintained (Jacquin and Shamseldin, 2009).

2.5 Support vector regression

2.5.1 Introduction

Support vector machine (SVM) is developed based on the statistical learning theory initiated by V.N. Vapnik in the 1970s at the Institute of Control Sciences of the Russian Academy of Science (Vapnik, 1998). Originally developed for classification (often referred to as support vector classification), it was extended to solve prediction problem (referred to as support vector regression (SVR)). Unlike most data-driven modeling methods where machine learning is to minimize difference between observed and predicted values (referred to as empirical risk minimization (ERM)), SVR implements structural risk minimization (SRM) that aims at minimizing a bound on the generalization error. In this way, it creates a model with a minimized

Vapnik–Chervonenkis (VC) dimension which means good generalization. Recently, SVR was used in hydrology-related tasks (Sivapragasam et al., 2001; Liong and Sivapragasam, 2002; Yu et al., 2006).

The function approximated by SVR can be linear and nonlinear depending on the simulated system being linear or nonlinear. Generally, a nonlinear SVR is adopted in hydrological forecasting since most issues of interest tend to be nonlinear. The nonlinear underlying function can be described as

$$f(\mathbf{X}, \mathbf{w}) = \mathbf{w}^T \cdot \Phi(\mathbf{X}) + b \quad (2.7)$$

where \mathbf{X} represents input vector, \mathbf{w} and b are the parameters of the function. $\Phi(\mathbf{X})$ is a nonlinear mapping function, which is used to map the input vector \mathbf{X} to a higher dimensional feature space in which the training data may exhibit linearity, and then to perform linear regression in this feature space. Eq. (2.7) stands for linear regression when $\Phi(\mathbf{X})$ is replaced by \mathbf{X} . The regression problem is actually an optimization problem. Its objective is to find optimal \mathbf{w} , b , and some parameters in $\Phi(\mathbf{X})$ to construct an approximation function of the underlying function. A wide range of literature can be referred to for the optimization in the context of the nonlinear SVR (Gunn, 1998; Dibike et al., 2001; Kecman, 2001; Sivapragasam et al., 2001; Liong and Sivapragasam, 2002; Cherkassky and Ma, 2004; Yu et al., 2006). Figure 2.4 illustrates the concept of nonlinear SVR, where only part input vectors contribute to the regression adopting a linear loss function having a Vapnik's ϵ -insensitivity tolerance error. Moreover, due to the tolerance, cost function is more robust against outliers and non-Gaussian noise of data. Parameter ξ_i and ξ_i^* are slack variables that are introduced to take into account outliers in the training data.

The parameter optimization in SVR can be solved in a dual space by introducing a dual set of Lagrange Multipliers, α_i and α_i^* . Substituting $\Phi(\mathbf{X})$ with appropriate kernel function the approximation function is expressed as

$$\hat{f}(\mathbf{X}_i) = \sum_{i=1}^N (\alpha_k - \alpha_k^*) K(\mathbf{X}_k \cdot \mathbf{X}_i) + b_0, \quad k = 1, \dots, n \quad (2.8)$$

Figure 2.5 illustrates the mapping relation in Eq. (2.8). Evidently, the mathematical

structure of SVR is extremely similar to that of ANN if the terms of $\alpha_k - \alpha_k^*$ and b_0 are regarded as weights and bias in the output layer. In essence, both SVR and ANN can be represented as two-layer networks (where the weights are non-linear in the first layer and linear in the second layer). A major distinction between the two approaches is the training algorithm. Moreover, while ANN generally adapts all the parameters, SVR chooses the parameters for the first layer to be the training input vectors because this minimizes the VC dimension (Cherkassky and Mulier 1998).

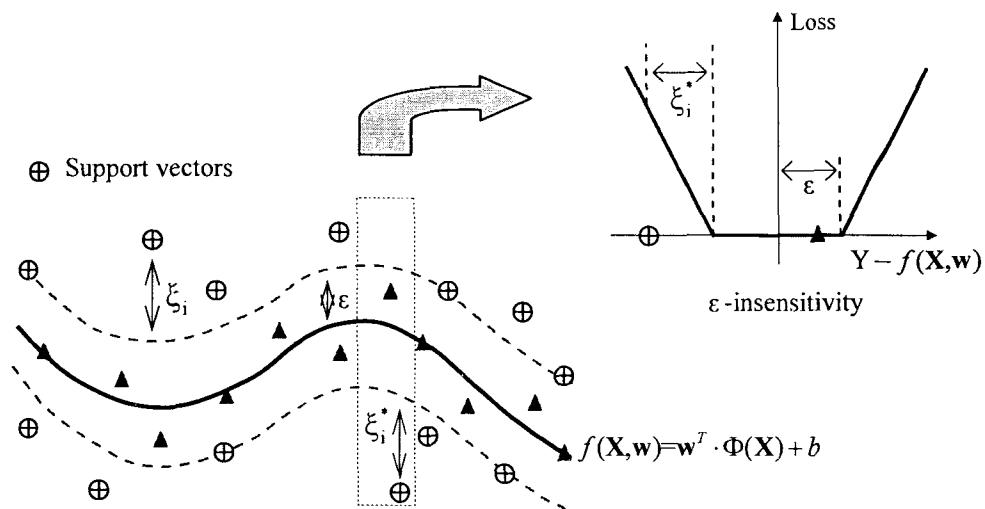


Figure 2.4 Nonlinear SVR with Vapnik's ϵ -insensitive loss function

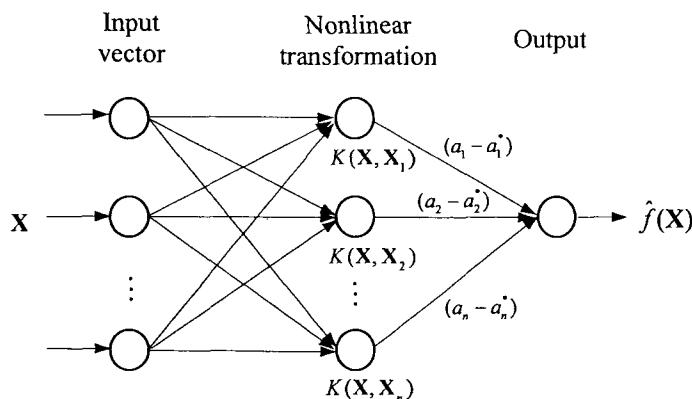


Figure 2.5 Configuration of SVR

2.5.2 Issues related to model identification

There is no fixed guideline for the identification of SVR model. According to a

practice guide to SV classification presented by Hsu et al. (2003), the identification of SVR model can be involved with inputs selection, data preprocessing, parameters calibration, and kernel function selection.

In view of the similarity between SVR and ANN, approaches for model inputs and data preprocessing are basically the same. For example, Liong and Sivapragasam (2002) used sensitivity analysis to find contributing stations to predict downstream flood stage. Sivapragasam and Liong (2005) determined model inputs by a trial and error procedure. Yu et al. (2006) selected input vector for real-time flood stage forecasting using a combination of statistical analysis and a prior knowledge on rainfall-runoff process. Moreover, the identification of appropriate rainfall lags is by a visual inspection of hyetograph and hydrograph. Lin et al. (2006) employed statistical analysis such as ACF and PACF to identify runoff series lags as model inputs.

Regarding data preprocessing, Sivapragasam et al. (2001) employed SSA to decompose rainfall and runoff data series. In order to equally treat input variables with different units receive same attention, some studies suggested rescaling original data to the interval $[-1, 1]$ or $[0, 1]$ (Hsu et al., 2003; Yu et al., 2006). Bray and Han (2004) investigated two different scaling schemes (one is to rescale training data by dividing with the maximum value, and the other is by dividing with two times of the maximum value), and results showed the former is better. In view of large data records which are not suitable to SVR, Yu et al. (2004) adapted a decomposition method to split data records in order to overcome the computational difficulty.

Commonly used kernel functions in SVR can be linear, polynomial, sigmoid, hyperbolic tangent, and radial basis function (Kecman, 2001). However, the preferred kernel function in hydrological forecasting is the Gaussian radial basis function (Liong and Sivapragasam, 2002; Hsu et al., 2003; Yu et al., 2004; Yu et al., 2006; Lin et al., 2006).

In the process of model training, the determination of C , ε and kernel-specific parameters are necessary. Some approaches have been reported in the literature. As is

well-known that the value of ε should be proportional to the input noise level, namely $\varepsilon \propto \sigma$ (Vapnik, 1999), Cherkassky and Ma (2004), from the perspective of analytic ‘rule-of-thumb’ selection, suggested selection of parameter C based on the formula of $\frac{C}{N} = \max(|\bar{x} + 3\sigma_x|, |\bar{x} - 3\sigma_x|)$, where \bar{x} and σ_x are the mean and the standard deviation of the training data x , and N is the data size of training data. ε is suggested as $\varepsilon = 3\sigma_a \sqrt{\ln N / N}$, where σ_a^2 is the variance of additive noise δ . In the meantime, global evolutionary techniques are widely employed for optimization of these parameters. Hsu et al. (2003) recommend a “grid-search” on C and parameter σ in RBF kernel using cross-validation. Lin et al. (2006) employed SCE-UA to search (C, ε, σ) . A two-step grid search method was applied to find the optimal parameter triplet (C, ε, σ) by Yu et al. (2006) and Wu et al. (2008). With the adoption of sigmoid function as kernel function in SVR, Yu et al. (2004) employed a shuffled complex evolution (SCE-UA) algorithm to search quintuplet parameters $(d, \tau, C, \varepsilon, \gamma)$ for chaotic runoff series forecasting.

2.5.3 Applications

As a newly emerging algorithm, the use of SVR in hydrology is not as widespread as that of other data-driven models such as ANN. Limited applications can be found in rainfall forecasting, river flow forecasting, and rainfall-runoff transformation. For example, Dibike et al. (2001) applied SVR to the rainfall-runoff forecasting, and results demonstrated that SVR outperforms ANN. Sivapragasam et al. (2001) reported application of SVR to predictions of rainfall and streamflow, where the input data are preprocessed by SSA. Liong and Sivapragasam (2002) applied SVR to river stage forecasting in Dhaka, Bangladesh, and concluded that SVR performs better than ANN in varied prediction leads. Yu et al. (2004) employed SVR to forecast a chaotic daily runoff series. Bray and Han (2004) applied SVR to the forecast of runoff, focusing on the identification of an appropriate model structure and relevant parameters. Sivapragasam and Liong (2005) first used sequential elimination approach to identify the optimal training set and then used SVR to forecast water stages. Lin et al. (2006) presented a SVR model to predict long-term

monthly flow discharge series, and SVR made a better accuracy of prediction compared with ARMA and ANN. Yu et al. (2006) carried out a real-time flood stage forecasting using SVR, where a two-step grid search method was used to optimize relevant parameters. In order to forecast a wide range of streamflow magnitude and improve prediction at large lead horizons, Sivapragasam and Liong (2005) proposed a modular SVR for streamflow forecasting in which three local SVRs are respectively used to simulate high, medium and low flows. The boundaries of flow categories were artificially set. Results showed the modular model was promising. Recently, fuzzy SVR was also reported in the work of Li and Chen (2005).

2.5.4 Advantages and disadvantages

SVR has some advantages over ANN although their structures are mathematically similar (e.g. Kecman, 2001; Sivapragasam et al., 2001; Liong and Sivapragasam, 2002; Bray and Han, 2004). To start with, SVR is not a black-box model (or a semi-black-box model) because the optimized Lagrange multipliers show the relative importance of the training patterns in arriving at the final decision. The parameter optimization in SVR is essentially a convex problem. Hence, there is only one optimum point, which means that the optimal structure of SVR is uniquely definite. On the contrary, ANN has many local optima, and finding the optimal structure of ANN is a time consuming and laborious task. In the meantime, outputs from ANN tend to be unstable for every time of training. In addition, multi-dimensional input vectors result in more complicated ANN configuration with more number of tunable parameters. In this regard, there is no increase in the number of tunable parameters of SVR with the increase of input dimension. Hence, SVR can handle multi-dimensional inputs more efficiently and easily. Finally, due to the implementation of SRM, SVR is robust against outliers and non-Gaussian noise of data (or good generation).

One of the main drawbacks of SVR is that it is usually computationally demanding and time-consuming. The computation time tends to exponentially increase with increase of the training data size. Nevertheless, correct kernel function is crucial to the success of SVR. Finally, like most of data-driven models, model extrapolation

can be poor since the model depends on the past records as support vectors.

2.6 Genetic programming

2.6.1 Introduction

Genetic programming (GP), which was developed by Koza (1992), has recently increased in popularity. The most frequently used GP method is so-called symbolic regression (Babovic and Keijzer, 2000). The aim of symbolic regression is to determine a functional relationship between input and output data sets. The resulting function may be linear, quadratic, or higher order polynomial, which is optimized by a randomized search method-usually a GA.

However, GP differs from traditional GA in that it typically operates on parse trees instead of bit strings (or termed lines of code). An example can be found in Figure 2.6, where the parse tree for the algebraic expression ' $a + (2.3 \times c)$ ' is shown. The root node is the first element of the tree, the interior nodes are the functions and the leaf nodes are the constants (e.g., 2.3) and/or the variables (e.g., a and c). If the set of functions used is sufficiently rich, tree structures are capable of representing hierarchical programs of any complexity. For example, these functions may include arithmetic operators ($+, -, \times, \div$), mathematical functions (sin, cos, log), Boolean operators (AND, OR, NOT), logical expressions (IF-THEN-ELSE), iterative functions (DO-UNTIL), or any other user-defined function. Whenever a node in a tree is created from the function set, a number of links equal to the number of arguments the function takes is created to radiate out from that node. The result of this process is a set of random trees (programs) of different sizes and shapes, each exhibiting a different fitness with respect to the objective function. Thus, the initial population is formed. Individual programs that best fit the data are then selected from the initial population. The programs that are the best fit are then selected to exchange part of the information between them to produce better programs through ‘crossover’ and “mutation”, which mimics the natural world’s reproduction process. Exchanging the parts of the best programs with each other is termed crossover, and randomly changing programs to create new programs is termed mutation. The programs that

fitted the data less well are discarded. This evolution process is repeated over successive generations and is driven towards finding symbolic expressions describing the data, which can be scientifically interpreted to derive knowledge about the process. Details on GP can be obtained from Koza (1992), Babovic and Abbott (1997), Babovic and Keijzer (2000), Khu et al. (2001), and Liong et al. (2002).

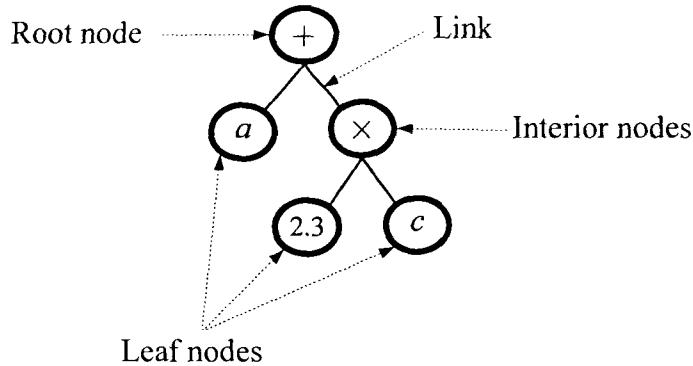


Figure 2.6 Tree representation of an expression

2.6.2 Applications

Genetic programming has been applied to an impressive range of problems. An overview of GP applications in hydrology can be found in Babovic and Keijzer (2005). Several representative examples can be mentioned. Savic et al. (1999) developed a GP approach to structured system identification for rainfall-runoff modeling. Khu et al. (2001) used GP to forecast real-time runoff. Babovic and Keijzer (2002) and Liong et al. (2002) applied GP to rainfall-runoff transformations. Babovic et al. (2003) employed GP to simulate velocity in compound channel with vegetated floodplain. Laucelli et al. (2007) present an application of GP to the problem of forecasting the groundwater heads in an aquifer in Italy and in this study the authors also conducted an ensemble forecasting built on the data subsets generated by bootstrap. Sivapragasam et al. (2007) compared GP and ANN for fortnightly flow series forecasting from one- to four-step lead using three types data as model inputs (namely, flow filtered by SSA, raw flow, and rainfall and flow). Results showed that the overall forecast performances of GP and ANN were similar in nature. For the perspective of model inputs, models with filtered flow data as inputs performed the best for short-term predictions. With the increase of the lead time, the forecast accuracy from models using filtered data as inputs deteriorates.

2.6.3 Advantages and disadvantages

When GP is compared to ANN, it exhibits some advantages (Giustolisi and Savic, 2006). GP generates a “transparent” and structured representation of the system being studied. Generally, the user has to predetermine the structure of the ANN network and the training algorithm, and only optimizes specific parameters of the network. However, GP do not require the identified model structure *a priori*. It can optimize both the structure of the model and the parameters. Moreover, ANNs do not provide direct relationships between the input and output variables as the relationship is contained in the connection weights, which is not accessible to human understanding at present (Savic et al. 1999). GP produces models that build an understandable structure, i.e., a formula or equation, which gives an insight into the relationship between input and output data. Besides, one of the advantages of genetic programming over more standard methods for regression is that an overall functional form need not be explicitly assumed. On the other hand, the GP method has some limitations, namely, GP is not very powerful in finding constants and, more importantly, it tends to produce more complex functions (and thus difficult to interpret) and carry no physical insight with an increase in prediction horizon (Davidson et al. 1999, 2000; Sivapragasam et al., 2007).

2.7 Comparison of modeling methods

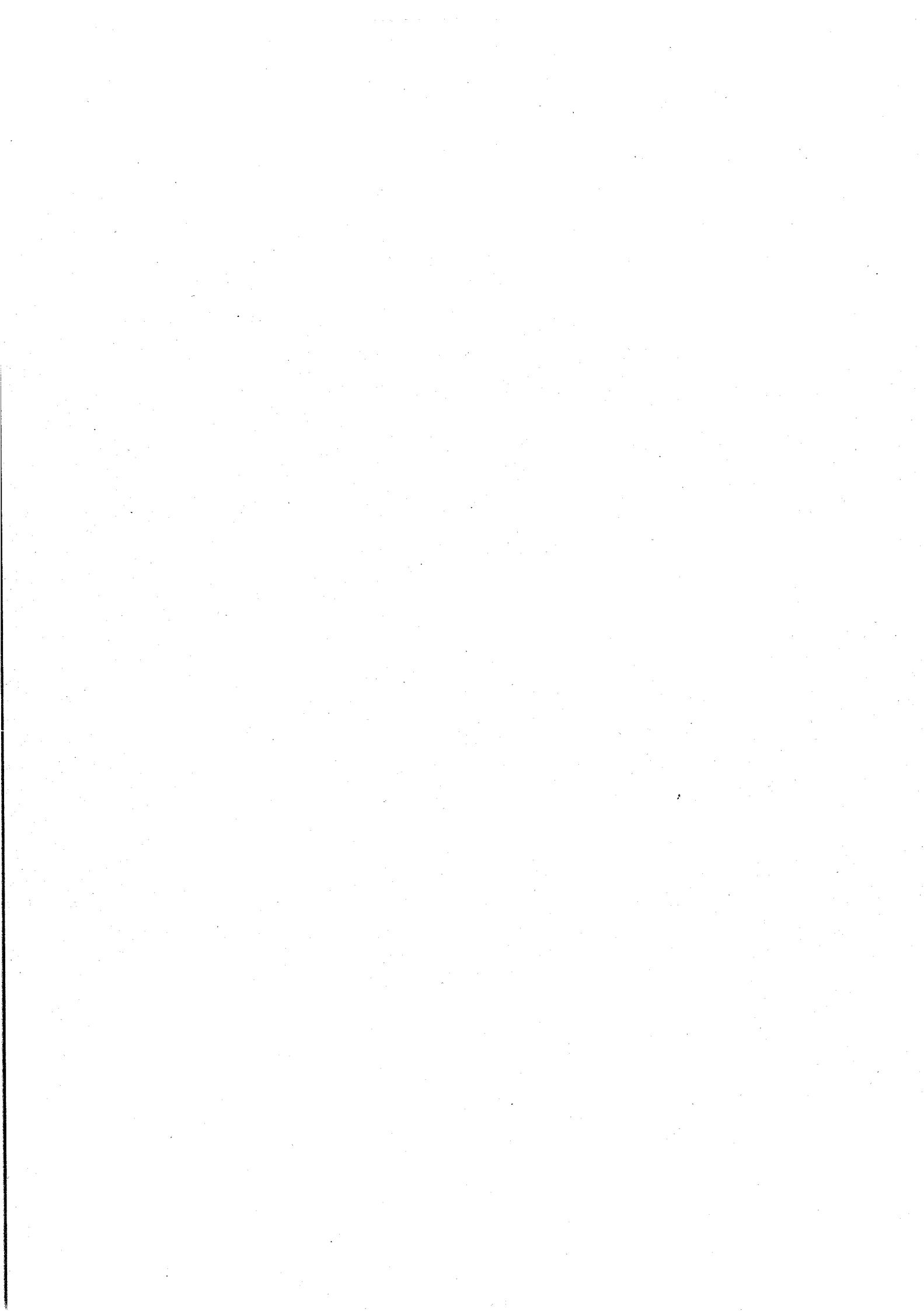
To clearly distinguish these modeling approaches, their merits and drawbacks mentioned above is summarized in Table 2.1. It is evident that there is no the optimal method and some of them tend to be mutually complementary. Therefore, developing a hybrid (or modular) model appears to be more useful than a global model.

Table 2.1 Brief summary on pros and cons of data-driven models

| Models | Pros | Cons |
|---|---|--|
| K-NN | <ul style="list-style-type: none"> · Be simple, but robust for noise and irrelevant attributes; · Be more transparent than ANN | <ul style="list-style-type: none"> ◊ Has no extrapolation ability; ◊ Be not suitable for real-time forecasting. |
| Chaos theory-based method | <ul style="list-style-type: none"> · Have accurate long term predictions if studied data series is chaos. | <ul style="list-style-type: none"> ◊ Be only suitable for chaotic hydrological series (no universal applicability). ◊ Has poor extrapolation ability. |
| ANN | <ul style="list-style-type: none"> · Can map any complex relationship at a high training speed; · Be very robust for the presence of noisy or incomplete data, and are able to deal with outliers; · Be well suited to longer-term forecasting; · Has a fast convergence with local optimization techniques. | <ul style="list-style-type: none"> ◊ Be not transparent and be not easily interpreted; ◊ Has a poor generalization and unstable model output; ◊ Be susceptible to local minima; ◊ Has no theoretic guidance for implementation of ANN; ◊ Be difficult in obtaining the optimal architecture; ◊ Requires a larger data set for training. ◊ Has poor extrapolation ability. |
| FIS | <ul style="list-style-type: none"> · Embed human (structured) knowledge into algorithms; · Approximate any multivariate nonlinear function; · Provide some insight about internal operation of models by IF-ELSE rule; · Can work with a small data; · Be appropriate tool in generic decision-making process; · Operate successfully under a lack of precise sensor information; | <ul style="list-style-type: none"> ◊ Learning is highly constrained and typically more complex than with ANNs; ◊ Has a slow convergence; ◊ Number of rules increases exponentially with increase of the number of input variables and their fuzzy subsets; ◊ Fuzzy rules may be affected by subjective knowledge of experts; ◊ Has poor extrapolation ability. |
| SVR | <ul style="list-style-type: none"> · Be not a complete black-box model; · Can approximate any multivariate nonlinear function; · Can obtain the optimal architecture as a solution of quadratic optimization problem; · Multi-dimensional inputs does not increase tunable parameters; · Has good generalization; · Be suitable for the situation of a small data set. | <ul style="list-style-type: none"> ◊ Be usually computationally intensive and time-consuming; ◊ Training time tends to increase exponentially with the number of training samples; ◊ Need to cautiously select parameters of C, ϵ and parameters in kernel function in order to avoid over fitting; ◊ Has poor extrapolation ability. |
| GP | <ul style="list-style-type: none"> · Be more transparent than ANN; · Model structure is usually understandable. | <ul style="list-style-type: none"> ◊ Model structure become not complex and carries no physical insight with an increase in prediction horizon, and are time consuming; ◊ Has poor extrapolation ability. |
| Comparison with classical regression models | <ul style="list-style-type: none"> · The training data do not have to follow a Gaussian distribution; · The data used may possess irregular seasonal variation; · Model is nonlinear and non-parametric; · Model's structure is unknown <i>a priori</i>. | <ul style="list-style-type: none"> ◊ Be computational intensive and time consuming. |

Part 2

Modeling



3 Determination of Model Inputs

Determining appropriate model inputs is extremely important when a data-driven model is developed to simulate a real system. If important candidate inputs are not included, some information about the system may be lost. On the contrary, an obtained data-driven model may misrepresent the system when spurious inputs are included.

As reviewed in Chapter 2, there are two categories of approaches for model input determination, either model-based (model dependent) or model-free. In model-based approaches, the input selection problem can be formulated as having a set of input variables to a model, and an output value, which can be used to evaluate the fitness or merit of the model using the input variables. There are typical methods such as trial-and-error (or stepwise) and sensitivity analysis. In model-free approaches, some statistical measures of dependence are used instead of using a model to characterize the relationship between input and output. If the dependence between input and output is found to be significant, the input is retained in the final input set. Otherwise, the input is discarded as it is unlikely to improve the predictive ability of a model. Some representative model-free methods include linear correlation analysis, mutual information analysis, and false nearest neighbors analysis.

Herein, seven commonly used approaches to determine model inputs are introduced, and they will be employed in later hydrological forecasts.

3.1 Linear Correlation Analysis

In the linear correlation analysis (LCA), the input vector corresponding to different antecedent observations generally depends on three statistical methods, i.e., cross correlation function (CCF), autocorrelation function (ACF) and partial autocorrelation function (PACF). Taking rainfall-runoff transformation as an

example, CCFs between rainfall and runoff are generally used to gather information about the lag of rainfall that heavily influences the runoff at a given time. ACF and PACF are used to gather information about the influencing lag runoff patterns in the flow at a certain time (Sudheer et al., 2002).

Suppose that we now have a finite time series $\{x_1, x_2, \dots, x_N\}$ of N observations. The estimate of ACF has appeared in many studies (Tsonis, 1992; Bowerman and O'Connell, 1987; Box et al., 1994). The most satisfactory estimate of the k th lag autocorrelation r_k is

$$r_k = \frac{c_k}{c_0} \quad (3.1)$$

where

$$c_k = \frac{1}{N} \sum_{t=1}^{N-k} (x_t - \bar{x})(x_{t+k} - \bar{x}) \quad k = 0, 1, 2, \dots \quad (3.2)$$

is the estimate of the autocovariance, and \bar{x} is the sample mean of the time series and is formulated as $\bar{x} = \frac{1}{N} \sum_{t=1}^N x_t$. Furthermore, substituting (3.2) into (3.1), r_k can be reformulated as

$$r_k = \frac{\sum_{t=1}^{N-k} (x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^N (x_t - \bar{x})^2} \quad (3.3)$$

According to Bowerman and O'Connell (1987), PACF can be calculated by the obtained ACF. Thus, the formula for the partial autocorrelation r_{kk} at lag k is

$$r_{kk} = \begin{cases} r_1 & \text{if } k = 1 \\ \frac{r_k - \sum_{t=1}^{k-1} r_{k-1,t} x_{k-t}}{1 - \sum_{t=1}^{k-1} r_{k-1,t} x_{k-t}} & \text{if } k = 2, 3, \dots \end{cases} \quad (3.4)$$

where

$$r_{kt} = r_{k-1,t} - r_{kk} x_{k-1} x_{k-t} \quad \text{for } t = 1, 2, 3, \dots, k-1 \quad (3.5)$$

To calculate the CCF between predictors and predictand, assume that there are N

pairs of observations $(x_1, y_1), (x_1, y_1), \dots, (x_N, y_N)$ available for analysis. Referring to Box et al. (1994), the estimate of cross-correlation $r_k(x, y)$ at lag k is

$$r_k(x, y) = \frac{c_k(x, y)}{s_x s_y} \quad (3.6)$$

where

$$c_k(x, y) = \begin{cases} \frac{1}{N} \sum_{t=1}^{N-k} (x_t - \bar{x})(y_{t+k} - \bar{y}) & k = 0, 1, 2, \dots \\ \frac{1}{N} \sum_{t=1}^{N+k} (y_t - \bar{y})(x_{t+k} - \bar{x}) & k = 0, -1, -2, \dots \end{cases} \quad (3.7)$$

is the estimate of the cross covariance coefficient, and \bar{x}, \bar{y} are the sample means of the x series and y series, respectively. s_x and s_y denote the estimates of standard deviations of x series and y series, respectively, and $s_x = \sqrt{c_0(xx)}$ and $s_y = \sqrt{c_0(yy)}$.

3.2 Average Mutual Information

Nonlinear relationships are commonly encountered in hydrological modeling and cannot be properly detected or quantified by linear correlation analysis between two variables. The mutual information (MI) criterion (Fraser and Swinney, 1986) is an attempt to capture all dependence (linear and/or nonlinear) between any two variables. Referring to Sharma (2000), Bowden et al. (2005), May et al. (2008), and Fernando et al. (2009), computation of MI is presented as follow. Given N pairs of observations $(x_1, y_1), (x_1, y_1), \dots, (x_N, y_N)$, the MI between two variables of \mathbf{X} and \mathbf{Y} , is defined as

$$I(\mathbf{X}; \mathbf{Y}) = \iint p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy \quad (3.8)$$

where $p(x)$ and $p(y)$ are the marginal probability density functions (PDFs) of \mathbf{X} and \mathbf{Y} , respectively, and $p(x, y)$ is the joint PDF of \mathbf{X} and \mathbf{Y} .

The rationale behind the MI function is the definition of dependence between the two variables. The joint probability of occurrence of the two variables is theoretically

equal to the product of the individual probabilities if there is no dependence between the variables. Hence, the joint probability density $p(x, y)$ would equal $p(x)p(y)$. The MI score in Eq. (3.8) would, in that case, equal a value of zero. A high value of the MI score would indicate a strong dependence between the two variables.

However, within a practical context, the true function forms of the PDFs in Eq. (3.8) are typically unknown. Hence, estimates of the densities are used instead. Substitution of density estimates into a numerical approximation of the integral in Eq. (3.8) gives

$$I(\mathbf{X}; \mathbf{Y}) = \frac{1}{N} \sum_{i=1}^N \log \frac{f(x_i, y_i)}{f(x_i)f(y_i)} \quad (3.9)$$

where $f(x_i)$, $f(y_i)$, and $f(x_i, y_i)$ are the respective univariate and joint probability densities estimated at the sample data points x_i and y_i . Note that the base of the logarithm varies within the literature and the use of either 2 or e was often reported. The natural logarithm (i.e. \ln) is assumed in this study, unless otherwise stated.

In hydrological modeling, average mutual information (AMI) is widely adopted, which is defined in bits as

$$\bar{I}(\mathbf{X}; \mathbf{Y}) = \sum_{x_i \in \mathbf{X}} \sum_{y_i \in \mathbf{Y}} f(x_i, y_i) \ln \frac{f(x_i, y_i)}{f(x_i)f(y_i)} \quad (3.10)$$

where $\bar{I}(\mathbf{X}; \mathbf{Y})$ denotes AMI, and the other symbols are the same as those mentioned above.

The key to an accurate estimate of the MI or AMI is the accurate estimation of the marginal and joint probability densities in Eq.(3.9) or Eq.(3.10). The original MI algorithms utilized crude measures to approximate the probability densities such as a histogram having a fixed bin width (Fraser and Swinney, 1986; Zheng and Billings, 1996). More recently, kernel density estimator techniques have been used due to their stability, efficiency, and robustness (Sharma, 2000).

Without loss of generality, the univariate set \mathbf{X} is extended to the d -dimensional variable set \mathbf{X} . The simple Parzen window forms the basis for the kernel density

estimator. The estimator is expressed as

$$\hat{f}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N K_h(\mathbf{x} - \mathbf{x}_i) \quad (3.11)$$

where $\hat{f}(\mathbf{x})$ is the multivariate kernel density estimate of the d -dimensional variable set \mathbf{X} at coordinate location \mathbf{x} ; \mathbf{x}_i the i th multivariate data point, for a sample of size N ; and K_h is some kernel function for which h denotes the kernel bandwidth (or smoothing parameter). A common choice for K_h is the Gaussian kernel expressed as

$$K_h = \frac{1}{(\sqrt{2\pi h})^d \sqrt{|\Sigma|}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|}{2h^2}\right) \quad (3.12)$$

where $\sqrt{|\Sigma|}$ is the sample covariance of the variable set \mathbf{X} , and $\|\mathbf{x} - \mathbf{x}_i\|$ is the Mahalanobis distance metric, which is given by

$$\|\mathbf{x} - \mathbf{x}_i\| = (\mathbf{x} - \mathbf{x}_i)^T \Sigma^{-1} (\mathbf{x} - \mathbf{x}_i) \quad (3.13)$$

Substituting the expression for the kernel into Eq. (3.11), the kernel density estimator becomes

$$\hat{f}(\mathbf{x}) = \frac{1}{N(\sqrt{2\pi h})^d \sqrt{|\Sigma|}} \sum_{i=1}^N \exp\left(-\frac{-(\mathbf{x} - \mathbf{x}_i)^T \Sigma^{-1} (\mathbf{x} - \mathbf{x}_i)}{2h^2}\right) \quad (3.14)$$

The bandwidth, h , is the key to an accurate estimate of the probability estimate of the probability density. Small value of h tends to lead to density estimate that gives too much emphasis to individual points. Large value of h , on the other hand, tends to over-smooth the probability density with all details, spurious or otherwise, becoming obscured. Various operational rules are available in the literature to help choose an optimal value of the bandwidth. Sharma (2000) used the Gaussian reference bandwidth (Scott, 1992) because it is relatively simple and computationally efficient

$$h_{\text{ref}} = \left(\frac{4}{d+2}\right)^{1/(d+4)} N^{(-1/(d+4))} \quad (3.15)$$

where N and d refer to the sample size and dimension of the multivariate variable set \mathbf{X} , respectively.

3.3 Partial Mutual Information

MI (or AMI) has recently been found to be a more suitable measure of dependence for input variable selection in data-driven modeling, since it is an arbitrary measure and makes no assumption regarding the structure of the dependence between variables. It is also robust due to its insensitivity to noise and data transformations (Battiti, 1994; Darbellay, 1999; Soofi and Retzer, 2003). However, several issues have arisen from the algorithm of MI or AMI. First of all, the algorithm cannot consider the inter-dependencies between candidate inputs, which may introduce redundant inputs to the input vector. Also, the algorithm lacks an appropriate analytical method for determining when the optimal set has been selected (Chow and Huang, 2005). Recently, Sharma (2000) proposed an input determination method based on the partial mutual information (PMI) criterion to overcome these difficulties.

Let \mathbf{X} and \mathbf{Z} represent two input variables and \mathbf{Y} be the output variable with the same sample size of N . If \mathbf{X} is a pre-existing predictor, the PMI between the dependent variable \mathbf{Y} and the independent variable \mathbf{Z} can be calculated as

$$\text{PMI}(\mathbf{Z}; \mathbf{Y} | \mathbf{X}) = I(\mathbf{V}; \mathbf{U}) \quad (3.16)$$

where

$$\mathbf{U} = \mathbf{Y} - \hat{m}_{\mathbf{Y}}(\mathbf{x}) \quad \mathbf{V} = \mathbf{Z} - \hat{m}_{\mathbf{Z}}(\mathbf{x}) \quad (3.17)$$

where $\hat{m}_{\mathbf{Y}}(\mathbf{x})$ and $\hat{m}_{\mathbf{Z}}(\mathbf{x})$ denote the regression estimators. Based on the kernel density estimation approach, the estimator $\hat{m}_{\mathbf{Y}}(\mathbf{x})$ for the regression of \mathbf{Y} on \mathbf{X} is expressed as

$$\hat{m}_{\mathbf{Y}}(\mathbf{x}) = E[y | \mathbf{X} = \mathbf{x}] = \frac{\sum_{i=1}^N y_i K_h(\mathbf{x} - \mathbf{x}_i)}{K_h(\mathbf{x} - \mathbf{x}_i)} \quad (3.18)$$

and the estimator $\hat{m}_{\mathbf{Z}}(\mathbf{x})$ for the regression of \mathbf{Z} on \mathbf{X} is written as

$$\hat{m}_{\mathbf{Z}}(\mathbf{x}) = E[z | \mathbf{X} = \mathbf{x}] = \frac{\sum_{i=1}^N z_i K_h(\mathbf{x} - \mathbf{x}_i)}{K_h(\mathbf{x} - \mathbf{x}_i)} \quad (3.19)$$

where K_h is the same as given in Eq. (3.12) and $E[\cdot]$ denotes the expectation operation.

The use of the conditional expectations in Eq. (3.17) ensures that the resulting variables \mathbf{U} and \mathbf{V} represent the residual information in variables \mathbf{Z} and \mathbf{Y} once the effect of the existing predictor \mathbf{X} has been taken into account.

As shown in Eq. (3.16), the computation of PMI is derived from MI. Both of PMI and MI require estimation of probability densities. The probability densities in MI are estimated for the original inputs and output whereas the probability densities in PMI are estimated for the residual information. The PMI-based input selection (PMIS) algorithm was originally developed by Sharma (2000) for the input identification of hydrological models. Given a candidate set, \mathbf{C} , and output variable \mathbf{Y} , the PMIS algorithm proceeds at each iteration by finding the candidate \mathbf{C}_s that maximizes the PMI with respect to the output variable, conditional on any previously selected inputs. The statistical significance of the PMI estimated for \mathbf{C}_s is assessed based on confidence bounds drawn from the distribution generation by a bootstrap loop. If the input is significant \mathbf{C}_s is added to \mathbf{S} (selected input set) and the selection continues; otherwise, there are no more significant candidates remaining and the algorithm is terminated. Detailed procedures of the PMIS algorithm are presented as follows (May et al., 2008).

- Let $\mathbf{S} \rightarrow \phi$ (Initialization)
- While $\mathbf{C} \neq \phi$ (Forward selection)
 - Construct kernel regression estimator $\hat{m}_{\mathbf{Y}}(\mathbf{s})$
 - Calculate residual output $\mathbf{U} = \mathbf{Y} - \hat{m}_{\mathbf{Y}}(\mathbf{s})$
 - For each $\mathbf{C}_j \in \mathbf{C}$
 - Construct kernel regression estimator $\mathbf{V} = \mathbf{C}_j - \hat{m}_{\mathbf{C}_j}(\mathbf{S})$
 - Estimate $I(\mathbf{V}; \mathbf{U})$
 - Find candidate \mathbf{C}_s (and \mathbf{V}_s) that maximizes $I(\mathbf{V}; \mathbf{U})$
 - For $b = 1$ to B (Bootstrap)
 - Randomly shuffle \mathbf{V}_s to obtain \mathbf{V}_s^*
 - Estimate $I_b = I(\mathbf{V}_s^*; \mathbf{U})$

- Find confidence bound $I_b^{(95)}$
- If $I(\mathbf{V}_s; \mathbf{U}) > I_b^{(95)}$ (Selection termination)
- Move \mathbf{C}_s to \mathbf{S}
- Else
- Break
- Return selected input set \mathbf{S}

In this procedure, B is the bootstrap size; and $I_b^{(95)}$ denotes the 95th percentile bootstrap estimate of the randomized PMI, I_b .

It is evident that PMI relies on a computationally intensive bootstrap estimation technique to implement an automatic termination criterion, which necessitates a trade-off between efficiency and accuracy of selection.

3.4 False Nearest Neighbors

The false-nearest-neighbors (FNN) algorithm was originally developed for determining the number of time-delay coordinates needed to recreate autonomous dynamics, but FNN has been extended to examine the problem of determining the proper embedding dimension m for input-output dynamics (Kennel et al., 1992; Abarbanel et al., 1993). Therefore, it is widely employed for the reconstruction of state space if a hydrological time series is treated as chaos. Let $\{x_1, x_2, \dots, x_N\}$ stands for a dynamic time series. It can be reconstructed into a series of delay vectors as $\mathbf{Y}_i = \{x_i, x_{i+\tau}, x_{i+2\tau}, \dots, x_{i+(m-1)\tau}\}$, $i = 1, 2, \dots, N - (m-1)\tau$, where $\mathbf{Y}_i \in \mathbb{R}^m$, τ is the delay time as a multiple of the sampling period and m is the embedding dimension. Suppose the point $\mathbf{Y}_j = \{x_j, x_{j+\tau}, x_{j+2\tau}, \dots, x_{j+(m-1)\tau}\}$ is identified in the Euclidean sense as the closest neighbor of a given point $\mathbf{Y}_i = \{x_i, x_{i+\tau}, x_{i+2\tau}, \dots, x_{i+(m-1)\tau}\}$, the criterion that \mathbf{Y}_j is viewed as a false neighbor of \mathbf{Y}_i is:

$$\frac{|x_{i+mt} - x_{j+mt}|}{\|\mathbf{Y}_i - \mathbf{Y}_j\|} > R_{tol} \quad (3.20)$$

where $\|\cdot\|$ stands for the distance in a Euclidean sense, R_{tol} is a threshold value with a common range of 10 to 30. For all points i in the vector state space, Eq. (3.20) is performed and then the percentage of points which have FNNs is calculated. The algorithm is repeated for increasing m until the percentage of FNNs drops to zero, or some acceptable small number such as 1%, where m is the target m .

3.5 Correlation Integral

The correlation integral is another method to identify the embedding dimension m for a hydrological time series characterized by chaos dynamics. The original formula of Grassberger-Procaccia algorithm (Grassberger and Procaccia, 1983) was modified by Theiler (1986) for the estimation of the correlation integral in a time series which poses the serious problem of temporal correlations. Thus, for a m -dimensional phase space, the modified correlation integral $C(r)$ is defined as

$$C(r) = \frac{2}{N_{pairs}} \sum_{i=1}^N \sum_{j=i+w+1}^{N-i} H(r - \|\mathbf{Y}_i - \mathbf{Y}_j\|) \quad (3.21)$$

where $N_{pairs} = (N - w + 1)(N - w)$, w is the Theiler window to exclude those points which are temporally correlated, \mathbf{Y}_i and N are given in the section of 3.4, r is the radius of a ball centered on \mathbf{Y}_i , H is the Heaviside step function, with $H(u) = 1$ if $u > 0$ and $H(u) = 0$ if $u \leq 0$. The correlation integral just counts the pairs $(\mathbf{Y}_i, \mathbf{Y}_j)$ whose distance (generally in a Euclidean sense) is smaller than r . In the limit of an infinite amount of data ($N \rightarrow \infty$) and sufficiently small r , the relation of $C(r) \propto r^{D_2}$ between $C(r)$ and r is expected when m exceeds the correlation dimension of the chaos theory, and the correlation exponent ν and correlation dimension D_2 can be respectively defined as $\nu = \partial \ln C(r) / \partial \ln r$ and $D_2 = \lim_{r \rightarrow 0} \nu$. Since one does not know D_2 before doing this computation, one checks for convergence of the

correlation dimension D_2 with respect to m .

The procedure of chaos identification is first to plot $\ln C(r)$ versus $\ln r$ with a given m , and then find the scaling region (if any) where the slope (i.e. the correlation exponent v) of the graph is approximately constant and is often estimated by a straight line fitting of the part of the graph. In general, the best way to find the scaling region is to produce a figure which shows the slope of the $\ln C(r)$ as a function of $\ln r$. The slope of a straight line is constant. Therefore, if a scaling region exists, then in a slope- $\ln r$ graph one should observe a plateau. This plateau provides an estimate for d_2 . It is worth noting that d_2 is the correlation dimension of the possible attractor for the present m . Repeating the procedure for successively higher m , if d_2 converges to a finite value D_2 (i.e. saturation value), this indicates a true attractor of dimension D_2 . The system under investigation may be considered as chaos. In the meantime, the m can be identified as the value that corresponds to the first occurrence of the saturation value D_2 in the plot of d_2 versus m .

However, an accurate estimation of d_2 requires a minimum number of points. Some researches claim that the size should be 10^A (Procaccia, 1988) or $10^{2+0.4m}$ (Tsonis, 1992), where A is the greatest integer smaller than d_2 and m ($m < 20$) is the embedding dimension used for estimating d_2 with an error less than 5%. Other research found that smaller data size is needed. For instance, the minimum data points for reliable d_2 is $10^{d_2/2}$ (Ruelle, 1990; Essex and Nerenberg, 1991), or $\sqrt{27.5}^{d_2}$ (Hong and Hong, 1994) and empirical results of dimension calculations are not substantially altered by going from 3000 or 6000 points to subsets of 500 points (Abraham et al., 1986).

3.6 Stepwise Linear Regression

The basic idea of the stepwise linear regression (SLR) is to start with a function that

contains the single best input variable and to subsequently add potential input variables to the function one at a time in an attempt to improve model performance. The order of addition is determined by using the partial F -test values to select which variable should enter next. The high partial F -value is compared to a (select or default) F -to-enter value. After a variable has been added, the function is examined to see if any variable should be deleted.

The basic procedure is described as follows according to Draper and Smith (1998). First, we select the X most correlated with Y (suppose it is X_1) and find the first-order, linear regression equation $\hat{Y} = f(X_1)$. We check if this variable is significant. If it is not, we quit and adopt the model $Y = \bar{Y}$ as best; otherwise we search for the second input variable to enter regression. We examine the partial F -values of all the input variables not in regression. The X_i with the highest such value (suppose it is X_2) is now selected and a second regression equation $\hat{Y} = f(X_1, X_2)$ is fitted. The overall regression is checked for significance, the improvement in the R^2 value (viz. the square of the correlation coefficient) is noted, and the partial F -values for two variables now in the equation are examined. The lower of these two partial F 's is then compared with an appropriate F percentage point, F -to-remove, and the corresponding input variable is retained in the equation or rejected according to whether the test is significant or not significant.

This testing of the “the least useful input currently in the equation” is carried out at every stage of the stepwise procedure. An input variable that may have been the best entry candidate at an earlier stage may, at a later stage, be superfluous because of the relationships between it and the other input variables in the regression. To check on this, this partial F criterion for each input variable in the regression at any stage of calculation is evaluated, and the lowest of these partial F -values is then compared with a preselected percentage point of the appropriate F -distribution or a corresponding default F -value. This provides a judgment on the contribution of the least valuable input variable in the regression at that stage, treated as though it had been the most recent variable entered, irrespective of its actual point of entry into the

model. If the tested input variable provides a non-significant contribution, it is removed from the model and the appropriate fitted regression equation is computed for all the remaining input variable still in the model.

The best of the input variables not currently in the model (viz. the one whose partial correlation with Y is greatest) is then checked to see if it passes the partial F entry test. If it passes, it is entered, and we return to checking all partial F 's for variables in. If it fails, a further removal is attempted. Eventually, when no variables in the current equation can be removed and the next best candidate variable cannot hold its place in the equation, the process stops.

3.7 ANN based on multi-objective genetic algorithm

The method of ANN based on multi-objective genetic algorithm (ANN^{MOGA}) was proposed for the design of the optimal ANN by Giustolisi and Simeone (2006), which attempts to overcome the curse of dimensionality and overfitting simultaneously. From the input selection standpoint, ANN^{MOGA} is a model-based method since obtaining the optimal inputs is accompanied by the optimization of the ANN's structure. Apart from the adoption of multi-objectives instead of single objective, the method is similar to GAGRNN developed by Bowden et al. (2005) which is a hybrid genetic algorithm and general regression neural network (GAGRNN).

According to Giustolisi and Simeone (2006), selection of both the model input and the number of hidden neurons should be performed concurrently because the best model input is interrelated to the number of hidden neurons and to the occurrence of overfitting. Evidently, this job is a combination problem because there are a great number of possible combinations (model input type and number of hidden neurons) in search of the best solution. For the sake of overcoming the curse of dimensionality and overfitting, authors employed three objective functions subject to minimization (i) one based on maximizing the fitness by minimizing the sum of square errors; (ii) the input dimension; and (iii) the number of hidden neurons. The adopted

multi-objective strategy is based on the Pareto dominance criterion allowing for the generation of the so-called Pareto front (Van Veldhuizen and Lamont, 2000) of non-dominated solutions. To solve the combinatorial optimization problem of discerning the Pareto model set, an evolutionary approach based on a MOGA strategy has been adopted.

With regard to the implementation of ANNs^{MOGA}, the ANNs^{MOGA} toolbox has been developed and can be freely downloaded at the website of www.hydroinformatics.it/. Some necessary descriptions pertaining to MOGA in the toolbox are shown as follows (Giustolisi and Simeone, 2006).

The decision variables (coded in the individual during evolution) were:

- a binary string for the model input components, first-layer bias and second-layer bias;
- two integer cells for coding the number of hidden neurons and decisions regarding the initial weights.

The objectives to minimize were:

- the choice of the function ($1 - \text{coefficient of determination}$) for computing fitness on the validation set;
- the dimension of the input (i.e. the number of components used in the model input); and
- the number of hidden neurons.

Specific features of MOGA are:

- selection based on ranking;
- multi-point crossover with probability rate equal to 0.4;
- single-point mutation with probability rate equal to 0.1;
- fixed population in evolution equal to 20 individuals; and
- Number of generation equals to 500.

4 Data Preprocessing Methods

Four techniques, namely, moving average (MA), principal component analysis (PCA), singular spectrum analysis (SSA), and wavelet analysis (WA), are often adopted to preprocess training data. These methods may improve the performance of a forecast model from different perspectives. For example, MA is used as a smoothing technique to mitigate the effect of irregular components and seasonal components. The use of PCA attempts to reduce the dimensionality of input variables and eliminate the multicollinearity (independent variables are interrelated). The purpose of SSA or WA is to improve the mapping ability between input and output data by removing potential noise components in the original training data. These techniques are used individually or jointly in this study.

4.1 Moving average

The MA method smoothes data by replacing each data point with the average of the k neighboring data points, where k may be termed the length of memory window. The method is based on this idea that any large irregular component at any point in time will exert a smaller effect if we average the point with its immediate neighbors (Newbold et al., 2003). The unweighted MA is the most commonly-used approach in which each value of the data carries the same weight in the smoothing process. There are three types of moving modes including centering, backward and forward. In a forecast scenario, the backward mode is only used since the other two modes necessitate future observed values. For a time series $\{x_1, x_2, \dots, x_N\}$, when the backward moving mode is adopted (Lee et al., 2000), the k -term unweighted moving average y_t^* is written as

$$y_t^* = \left(\sum_{i=0}^{k-1} y_{t-i} \right) / k \quad (4.1)$$

where $t = k, \dots, N$. Choice of the window length k is by a trial and error procedure of minimizing the loss of the objective function.

4.2 Principal Component Analysis

PCA was firstly introduced by Pearson (1901) and developed independently by Hotelling (1933), and has now well entrenched as an important technique in data analysis. The central idea is to reduce the dimensionality of a data set consisting of a large number of interrelated variables, while retaining as much as possible of the variation present in the data set. The PCA approach uses all of the original variables to obtain a smaller set of new variables (principal components-PCs) which can be used to approximate the original variables. PCs are uncorrelated and are ordered so that the first few retain most of the variation present in the original set.

Consider a vector \mathbf{x} with p variables of x_1, x_2, \dots, x_p . Let the covariance matrix of \mathbf{x} be Σ . The purpose of PCA is to determine a new variable z that can be used to account for the variation in the p variables x_1, x_2, \dots, x_p . For example, the first PC is given by a linear combination of the p variables as

$$z_1 = \mathbf{a}_1^T \mathbf{x} = a_{11}x_1 + a_{12}x_2 + \dots + a_{1p}x_p \quad (4.2)$$

where \mathbf{a}_1^T is the transpose of the vector \mathbf{a}_1 consisting of coefficient $a_{11}, a_{12}, \dots, a_{1p}$.

The vector \mathbf{e}_1 maximizes $\text{var}[\mathbf{a}_1^T \mathbf{x}] = \mathbf{a}_1^T \Sigma \mathbf{a}_1$ subject to $\mathbf{a}_1^T \mathbf{a}_1 = 1$. By introducing Lagrange multiplier λ_1 , the maximization problem turns into solving the equation

$$(\Sigma - \lambda_1 \mathbf{I}_p) \mathbf{a}_1 = 0 \quad (4.3)$$

where \mathbf{I}_p is the $(p \times p)$ identity matrix. Thus, λ_1 is an eigenvalue of Σ and \mathbf{e}_1 is the corresponding eigenvector. Due to the requirement of maximum variance, the λ_1 must be as large as possible. Hence, it is noted that λ_1 is the largest eigenvalue of Σ and \mathbf{a}_1 is the corresponding eigenvector. Similar equations to Eq. (4.3) can be established for PCs from z_2 to z_p (namely, $(\Sigma - \lambda_i \mathbf{I}_p) \mathbf{a}_i = 0$ where $i = 2, \dots, p$). The vectors of coefficients $\mathbf{a}_2, \mathbf{a}_3, \dots, \mathbf{a}_p$ are the eigenvectors of Σ corresponding to $\lambda_2, \lambda_3, \dots, \lambda_p$ which is in descending order of their values.

PCA is scale dependent, and so the data must be scaled in some meaningful way. The most usual way of scaling is to scale each variable to unit variance.

Now extend a vector \mathbf{x} to a data matrix \mathbf{X} which has n rows (observations) and p column (variables). Let the covariance matrix of \mathbf{X} be Σ , where $\Sigma = \text{cov}(\mathbf{X}) = E(\mathbf{X}^T \mathbf{X})$. The linear transformed orthogonal matrix \mathbf{Z} is presented as

$$\mathbf{Z} = \mathbf{XA} \quad (4.4)$$

where \mathbf{Z} is the PCs with elements (i, j) of i th observation and j th principal component; \mathbf{A} is a $(p \times p)$ matrix with eigenvector elements of the covariance of \mathbf{X} , and having $\mathbf{A}^T \mathbf{A} = \mathbf{AA}^T = \mathbf{I}$.

Because matrix $\mathbf{X}^T \mathbf{X}$ is real and symmetric, it can be expressed as $\mathbf{X}^T \mathbf{X} = \mathbf{A} \Lambda \mathbf{A}^T$ where Λ is a diagonal matrix whose nonnegative entries are the eigenvalues $(\lambda_i, i=1, \dots, p)$ of $\mathbf{X}^T \mathbf{X}$. The total variance of the data matrix \mathbf{X} is represented as

$$\text{trace}(\Sigma) = \text{trace}(\mathbf{A} \Lambda \mathbf{A}^T) = \text{trace}(\Lambda) = \sum_{i=1}^p \lambda_i \quad (4.5)$$

On the other hand, the covariance matrix of principal components \mathbf{Z} is expressed as

$$\text{cov}(\mathbf{Z}) = E(\mathbf{Z}^T \mathbf{Z}) = E(\mathbf{A}^T \mathbf{X}^T \mathbf{X} \mathbf{A}) = \Lambda \quad (4.6)$$

$$\text{trace}(\mathbf{Z}) = \text{trace}(\Lambda) = \sum_{i=1}^p \lambda_i \quad (4.7)$$

Therefore, the total variance of the data matrix \mathbf{X} is identical to the total variance after PCA transformation \mathbf{Z} .

The solution of PCA, using singular value decomposition (SVD) or determinants of the covariance matrix of \mathbf{X} , can provide the eigenvectors \mathbf{A} with their eigenvalues, $\lambda_i, i=1, \dots, p$, representing the variance of each component after PCA transformation. If eigenvalues are ordered by $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_p \geq 0$, the first few PCs can capture most of the variance of the original data while the remaining PCs mainly represent noise in the data. The percentage of total variance explained by the first m th PCs is

$$V = \sum_{i=1}^m \lambda_i / \sum_{i=1}^p \lambda_i \cdot 100\% \quad (4.8)$$

The higher the selection of the total data variance, V , the better the properties of the data matrix are preserved. For the sake of the reduction of dimensionality, a small number of PCs are selected, but still remain most of the data variance in selected components. If the transformation is to prevent the collinearity of regression variables, the selected component number m in Eq. (4.8) can be set for a higher total variance, such as $V = 95\% \sim 99\%$ (Hsu et al., 2002).

The original data matrix \mathbf{A} can be reconstructed by a reverse operation of Eq. (4.4) as

$$\mathbf{X} = \mathbf{ZA}^T \quad (4.9)$$

By choosing suitable $m (\leq p)$ PCs from \mathbf{Z} and accompanying m eigenvectors from \mathbf{A} , the original data can be filtered.

4.3 Singular Spectrum Analysis

SSA is a novel technique for analyzing time series incorporating the elements of classical time series, multivariate statistics, multivariate geometry, dynamical systems and signal processing. Its aim is to decompose the original series into a sum of a small number of interpretable components such as a slowly varying trend, oscillatory components and a “structureless” noise (Golyandina et al. 2001). Based on these components, it also provides prediction models (Vautard et al., 1992). SSA, as a data preprocessing technique, is of major concern in the current study. In this regard, SSA is used to perform a spectrum analysis on the input data, eliminate the “irrelevant features” (high-frequency components or noise) and invert the remaining components to yield a “filtered” time series. This approach of filtering a time series to retain desired modes of variability is based on this idea that the predictability of a system can be improved by forecasting the important oscillations in time series taken from the system. SSA has been used as an efficient preprocessing algorithm coupled with neural networks (or similar approaches) for time series forecasting (Lisi et al., 1995; Sivapragasam et al., 2001; Baratta et al., 2003; Sivapragasam et al., 2007). For example, Lisi et al. (1995) applied SSA to extract the significant components in their study on Southern Oscillation Index (SOI) time series and used ANN for prediction.

They reconstructed the original series by summing up the first “p” significant components. Sivapragasam et al. (2007) employed GP and ANN coupled with SSA for river flow forecasting. Results showed that SSA can significantly improve model performance in short prediction horizons. Two types of SSA, the Toeplitz SSA (Vautard and Ghill, 1989) and the basic SSA (Golyandina et al., 2001), are usually employed in hydrological forecasting. The Toeplitz SSA is a well known modification of the basic SSA.

4.3.1 Basic SSA

According to Golyandina et al. (2001), the basic SSA consists of two stages: decomposition and reconstruction. The decomposition stage involves two steps: embedding and SVD; the reconstruction stage also comprises two steps: grouping and diagonal averaging. Consider a real-valued time series $F = \{x_1, x_2, \dots, x_N\}$ of length $N(> 2)$. Assume that the series is a nonzero series, viz. there exists at least one i such that $x_i \neq 0$. Four steps are briefly presented as follows.

1st step: embedding

The embedding procedure maps the original time series to a sequence of multi-dimensional lagged vectors. Let L be an integer (window length), $1 < L < N$, and τ be the delayed time as the multiple of the sampling period. The embedding procedure forms $n (= N - (L-1)\tau)$ lagged vectors $\mathbf{x}_i = \{x_i, x_{i+\tau}, x_{i+2\tau}, \dots, x_{i+(L-1)\tau}\}^T$, where $\mathbf{x}_i \in \mathbb{R}^L$, and $i = 1, 2, \dots, n$. The ‘trajectory matrix’ of the time series is denoted by $\mathbf{X} = [\mathbf{x}_1 \ \dots \ \mathbf{x}_i \ \dots \ \mathbf{x}_n]$ having lagged vectors as its columns. In other words, the trajectory matrix is

$$\mathbf{X} = \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ x_{1+\tau} & x_{2+\tau} & x_{3+\tau} & \dots & x_{n+\tau} \\ x_{1+2\tau} & x_{2+2\tau} & x_{3+2\tau} & \dots & x_{n+2\tau} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{1+(L-1)\tau} & x_{2+(L-1)\tau} & x_{3+(L-1)\tau} & \dots & x_N \end{pmatrix} \quad (4.10)$$

If $\tau = 1$, the matrix \mathbf{X} is termed Hankel matrix since it has equal elements on the

‘diagonals’ where the sum of subscripts of row and column is equal to constant. If $\tau > 1$, the equal elements in \mathbf{X} are not definitely in the ‘diagonals’.

2nd step: SVD

Let $\mathbf{S} = \mathbf{XX}^T$. Denoted by $\lambda_1, \lambda_2, \dots, \lambda_L$ the eigenvalues of \mathbf{S} taken in the decreasing order of magnitude ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_L \geq 0$) and by $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_L$ the orthonormal system of the eigenvectors of the matrix \mathbf{S} corresponding to these eigenvalues. If we denote $\mathbf{v}_i = \mathbf{X}_i^T \mathbf{u}_i / \sqrt{\lambda_i}$ ($i = 1, \dots, L$) (equivalent to the i th eigenvector of $\mathbf{X}^T \mathbf{X}$), then the SVD of the trajectory matrix \mathbf{X} can be written as

$$\mathbf{X} = \mathbf{X}_1 + \dots + \mathbf{X}_L \quad (4.11)$$

where $\mathbf{X}_i = \sqrt{\lambda_i} \mathbf{u}_i \mathbf{v}_i^T$. The matrices \mathbf{X}_i have rank 1; therefore they are elementary matrices. The collection $(\lambda_i, \mathbf{u}_i, \mathbf{v}_i)$ will be termed i th eigentriple of the SVD. Note that \mathbf{u}_i and \mathbf{v}_i are also i th left and right singular vectors of \mathbf{X} , respectively.

3rd step: grouping

The purpose of this step is to appropriately identify the trend component, oscillatory components with different periods, and structureless noises by grouping components. This step can be also skipped if one does not want to precisely extract hidden information by regrouping and filter of components.

The grouping procedure partitions the set of indices $\{1, \dots, L\}$ into m disjoint subsets I_1, \dots, I_m , so the elementary matrix in Eq. (4.11) is regrouped into m groups. Let $I = \{i_1, \dots, i_p\}$. Then the resultant matrix \mathbf{X}_I corresponding to the group I is defined as $\mathbf{X}_I = \mathbf{X}_{i_1} + \dots + \mathbf{X}_{i_p}$. These matrices are computed for I_1, \dots, I_m and substituting into the expansion (4.11) one obtains the new expansion

$$\mathbf{X} = \mathbf{X}_{I_1} + \dots + \mathbf{X}_{I_m} \quad (4.12)$$

The procedure of choosing the sets I_1, \dots, I_m is termed the eigentriple grouping.

4th step: Diagonal averaging

The last step in the Basis SSA transforms each resultant matrix of the grouped decomposition (4.12) into a new series of length N . The diagonal averaging is to find equal elements in the resultant matrix and then to generate a new element by averaging over them. The new element has the same position (or index) as that of these equal elements in the original series. As mentioned in the step 1, the concept of ‘diagonal’ is not true for $\tau > 1$. Regardless of the value of τ larger than or equal 1, the principle of reconstruction is the same. For $\tau = 1$, the diagonal averaging can be carried out by formula recommended by Golyandina et al. (2001). Let \mathbf{Y} be a $(L \times n)$ matrix with elements y_{ij} , $1 \leq i \leq L$, $1 \leq j \leq n$. Make $L^* = \min(L, n)$, $n^* = \max(L, n)$ and $N = n + (L - 1)\tau$. Let $y_{ij}^* = y_{ij}$ if $L < n$ and $y_{ij}^* = y_{ji}$ otherwise. Diagonal averaging transfers matrix \mathbf{Y} to a series $\{y_1, y_2, \dots, y_N\}$ by the formula

$$y_k = \begin{cases} \frac{1}{k} \sum_{m=1}^k y_{m,k-m+1}^* & \text{for } 1 \leq k < L^* \\ \frac{1}{L^*} \sum_{m=1}^{L^*} y_{m,k-m+1}^* & \text{for } L^* \leq k \leq K^* \\ \frac{1}{N-k+1} \sum_{m=k-K^*+1}^{N-K^*+1} y_{m,k-m+1}^* & \text{for } L^* < k \leq N \end{cases} \quad (4.13)$$

Eq. (4.13) corresponds to averaging of the matrix elements over the ‘diagonals’ $i + j = k + 1$. The diagonal averaging, applied to a resultant matrix \mathbf{X}_{I_k} , produces a N -length series F_k , and thus the original series F is decomposed into the sum of m series:

$$F = F_1 + \dots + F_m \quad (4.14)$$

As mentioned above, these reconstructed components (RCs) can be associated with the trend, oscillations or noise of the original time series with proper choices of L and the sets of I_1, \dots, I_m . Certainly, if the third step (namely, grouping) is skipped, F can be decomposed into L RCs.

4.3.2 Toeplitz SSA

The Toeplitz SSA was suggested by Vautard and Ghill (1989) and it is a well known modification of the basic SSA and is considered in Golyandina et al. (2001) as a supplementary SSA technique. It is based on particular non-optimal decompositions of the trajectory matrices, and may be useful in analysis of time series of special structure, such as series with linear-like tendencies and stationary-like series. A preliminary introduction to this SSA can be found in Vautard et al. (1992) and Elsner and Tsonis (1996). According to their work, four steps are summarized for the implementation of the Toeplitz SSA.

The first step is to construct the “trajectory matrix”. The “trajectory matrix” results from the method of delays. Consider a time series $F = \{x_1, x_2, \dots, x_N\}$, the “trajectory matrix” is denoted by

$$\mathbf{X} = \frac{1}{\sqrt{N}} \begin{pmatrix} x_1 & x_{1+\tau} & x_{1+2\tau} & \dots & x_{1+(L-1)\tau} \\ x_2 & x_{2+\tau} & x_{2+2\tau} & \dots & x_{2+(L-1)\tau} \\ x_3 & x_{3+\tau} & x_{3+2\tau} & \dots & x_{3+(L-1)\tau} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & x_{n+\tau} & x_{n+2\tau} & \dots & x_N \end{pmatrix} \quad (4.15)$$

where L is the embedding dimension (also termed singular number in the context of SSA), τ is the lagged (or delay) time. The matrix dimension is $n \times L$ where $n = N - (L - 1)\tau$.

The next step is the SVD of the trajectory matrix \mathbf{X} . Let $\mathbf{S} = \mathbf{X}^T \mathbf{X}$ (termed lagged-covariance matrix or Toeplitz matrix). With SVD, \mathbf{X} can be written as $\mathbf{X} = \mathbf{D} \mathbf{L} \mathbf{E}^T$ where \mathbf{D} and \mathbf{E} are left and right singular vectors of \mathbf{X} , and \mathbf{L} is a diagonal matrix of singular values. \mathbf{E} consists of orthonormal columns, and is also termed the ‘empirical orthonormal functions’ (EOFs). Substituting \mathbf{X} into the definition of \mathbf{S} yields the formula of $\mathbf{S} = \mathbf{E} \mathbf{L}^2 \mathbf{E}^T$. Further $\mathbf{S} = \mathbf{E} \wedge \mathbf{E}^T$ since $\mathbf{L}^2 = \wedge$ where \wedge is a diagonal matrix consisting of ordered values $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_L \geq 0$. Therefore, the right singular vectors of \mathbf{X} are the eigenvectors of \mathbf{S} (Elsner and Tsonis, 1996). In other words, the singular vectors \mathbf{E} and singular values of \mathbf{X} can be respectively attained by calculating the eigenvectors and the square roots of

the eigenvalues of \mathbf{S} .

Once \mathbf{E} is obtained, the subsequent third step is to calculate the principal components (a_i^k , $k = 1, \dots, L$) by projecting the original time record onto the eigenvectors as follows:

$$a_i^k = \sum_{j=1}^L x_{i+(j-1)\tau} e_j^k \quad \text{for } i = 1, 2, \dots, n \quad (4.16)$$

where e_j^k represents the j th component of the k th eigenvector in \mathbf{E} .

The last step is to generate RCs whose lengths are the same as the original series. The generation of each RC depends on a convolution of one principal component with the corresponding singular vector, which also necessitates the diagonal averaging operation. For $\tau = 1$, the formula of RC is given by Vautard et al. (1992) as follows

$$(R_A x)_i = \begin{cases} \frac{1}{L} \sum_{j=1}^L \sum_{k \in A} a_{i-j}^k e_j^k & \text{for } L \leq i \leq N - L + 1 \\ \frac{1}{i} \sum_{j=1}^i \sum_{k \in A} a_{i-j}^k e_j^k & \text{for } 1 \leq i \leq L - 1 \\ \frac{1}{N-i+1} \sum_{j=i-N+L}^L \sum_{k \in A} a_{i-j}^k e_j^k & \text{for } N - L + 1 < i \leq N \end{cases} \quad (4.17)$$

where A is the set including different indices of k . When A consists of a single index k , the series $R_A x$ is termed the k th RC, and will be denoted by x^k . RCs have additive properties, i.e.,

$$R_A x = \sum_{k \in A} x^k \quad (4.18)$$

In particular, the original time series can be completely restored when $A = \{1, \dots, L\}$

$$F = \sum_{k=1}^L x^k \quad (4.19)$$

Therefore, SSA can also perform the filtering and extraction of characteristic signals when the set A is appropriately set by selecting indices from $\{1, \dots, L\}$.

For $\tau > 1$, for the sake of reconstruction, two or three more steps may be needed,

which is identical to relevant steps for $\tau > 1$ in the basic SSA. Firstly, the elementary matrix ($n \times L$) has to be generated by the formula

$$\mathbf{X}^k = a^k (e^k)^T \quad (4.20)$$

The SVD of the trajectory matrix \mathbf{X} can thus be written as

$$\mathbf{X} = \mathbf{X}^1 + \cdots + \mathbf{X}^L \quad (4.21)$$

If there is no need to extract components, the grouping operation is skipped. Finally, the diagonal averaging is applied to each resultant matrix for the generation of RCs.

It should be noted that the two SSA techniques yield almost the same RCs although the operation procedure is slightly different, which has been found in the present forecasting experiments. Therefore, only one of them is adopted in the later applications.

4.4 Wavelet Analysis

The WA method in this study aims at utilizing discrete wavelet transform (DWT) to decompose a raw signal into a series of component signals. These components consist of one containing its trend (approximation) and others containing the high-frequency events (details). Referring to Daubechies (1992) and Küçük and Ağiralıoğlu (2006), DWT is briefly presented as follows together with an introduction of continuous wavelet transform (CWT) since it is the basis of DWT.

4.4.1 CWT

Let $f(t)$ be a continuous time series with $t \in [-\infty, \infty]$, the CWT of $f(t)$ with respect to a wavelet function $\psi(t)$ is defined by the linear integral operator

$$W(a, b) = \int_{-\infty}^{\infty} f(t) \psi_{a,b}^*(t) dt \quad (4.22)$$

where

$$\psi_{a,b}^*(t) = \frac{1}{\sqrt{|a|}} \psi^* \left(\frac{t-b}{a} \right) \quad (4.23)$$

where $W(a,b)$ is the wavelet coefficients and a and b are real numbers; (*) indicates complex conjugation. Thus, the wavelet transform is a function of two variables, a and b . The parameter ‘ a ’ can be interpreted as a dilation ($a>1$) or contraction ($a<1$) factor of the wavelet function $\psi(t)$ corresponding to different scales of observation. The parameter ‘ b ’ can be interpreted as a temporal translation or shift of the function $\psi(t)$, which allows the study of the signal $f(t)$ locally around the time b . The wavelet transform therefore expresses a time series in three-dimensional space: time (b), scale/frequency (a), and wavelet spectrum $|W(a,b)|^2$. CWT provides a time-frequency representation of a time series. The wavelet spectrum can also be averaged in time, referred to as the global wavelet power spectrum, allowing the determination of the characteristic scales and the characteristic periods of oscillation. The free software for CWT provided by Torrence and Campo (1998) can be downloaded at the website of <http://paos.colorado.edu/research/wavelets/>.

4.4.2 DWT

DWT is to calculate the wavelet coefficients on discrete dyadic scales and positions in time. Discrete wavelet functions have the form by choosing $a = a_0^m$ and $b = nb_0a_0^m$ in Eq. (4.23) as:

$$\psi_{m,n}(t) = a_0^{-m/2} \psi\left(\frac{t - nb_0a_0^m}{a_0^m}\right) = a_0^{-m/2} \psi(a_0^{-m}x - nb_0) \quad (4.24)$$

where m and n are integers that control the wavelet dilation and shift respectively, and $a_0 > 1$, $b_0 > 0$ are fixed. The appropriate choices for a_0 and b_0 depend on the wavelet function. A common choice for them is $a_0 = 2$, $b_0 = 1$. Now Assuming a discrete time series x_i , where x_i occurs at the discrete time i , DWT becomes

$$W_{m,n} = 2^{-m/2} \sum_{i=0}^{N-1} x_i \psi(2^{-m}i - n) \quad (4.25)$$

where $W_{m,n}$ is the wavelet coefficient for the discrete wavelet function with scale $a = 2^m$ and location $b = 2^m n$. In this study, the wavelet function (or wavelet base) is derived from the family of Daubechies wavelets with the 3 order.

4.4.3 Multi-resolution analysis (MRA)

The Mallat's decomposition algorithm (Mallat, 1989) is employed in this study. According to the Mallat's theory, the original discrete time series x_i is decomposed into a series of linearly independent approximation and detail signals.

The process consists of a number of successive filtering steps as depicted in Figure 4.1. Figure 4.1 (a) displays an entire MRA scheme, and Figure 4.1 (b) shows the filtering operation between two adjacent resolutions. The original signal x_i is first decomposed into an approximation and an accompanying detail. The decomposition process is then iterated, with successive approximation being decomposed in turn so that the finest-resolution original signal is transformed into many coarser-resolution components (Küçük and Ağaloğlu, 2006). As shown in Figure 4.1 (b), the approximation cA_{i+1} is achieved by letting cA_i pass through the low-pass filter H' and downsampling by two (denoted as $\downarrow 2$) whereas the detailed version cD_{i+1} is obtained by letting cA_i pass through the high-pass filter G' and downsampling by two. The details are therefore the low-scale, high frequency components whereas the approximations are the high-scale, low-frequency components. Finally, the original signal x_i is decomposed into many detailed components and one approximation component which may reflect a trend in the raw series. Following the procedure, the raw flow data can be decomposed into $m+1$ components if the m in DWA is set.

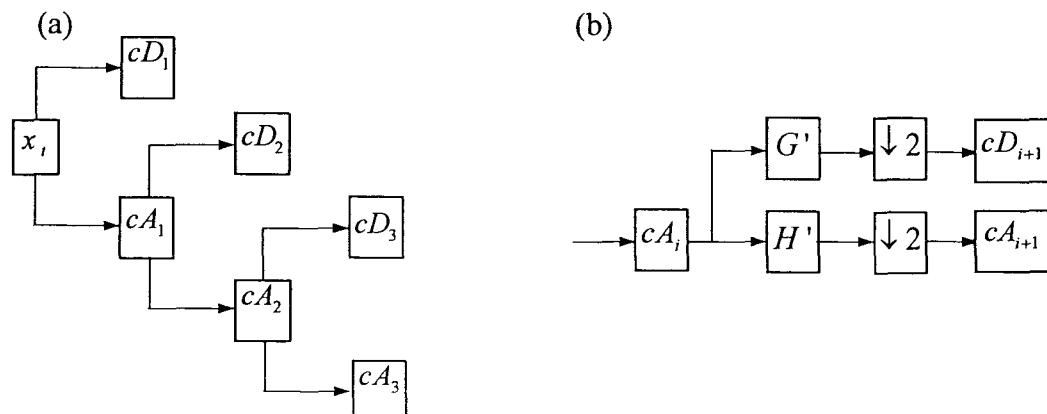


Figure 4.1 Schematics of (a) decomposition of x_i at level 3 and (b) signal filter

5 Potential Forecasting Models

Model input and data preprocessing have been introduced in Chapters 3 and 4, respectively. The ensuing task is to develop appropriate models for rainfall and streamflow predictions. Several representative modeling techniques including LR, K-NN, DSBM, ANN, SVR, MANN, and hybrid of ANN and SVR, will be employed in Chapters 6, 7, and 8. K-NN and DSBM (specially referring to as the chaos theory-based method) have been described in Chapter 2. Herein, descriptions of forecasting models concentrate on ANN, SVR, and modular (or hybrid) models of them. Moreover, combinations of models with data preprocessing techniques are also elucidated.

5.1 Artificial Neural Networks

As mentioned in Chapter 2, the MLP network is by far the most popular ANN paradigms, which usually uses the technique of error back propagation (BP) to train the network configuration. The architecture of ANN consists of the number of hidden layers and the number of neurons in the input layer, hidden layers and output layer. ANNs with one hidden layer are commonly used in hydrologic modeling (Dawson and Wilby, 2001; de Vos and Rientjes, 2005) since these networks are considered to provide enough complexity to accurately simulate the nonlinear-properties of the hydrologic process. Take a univariate time series as an example, a three-layer (viz. $m-h-1$) ANN forecasting model is described as follows. Let the time series be $\{x_1, x_2, \dots, x_N\}$. Based on the delay method, a set of vector $\mathbf{Y}_t = \{x_t, x_{t+\tau}, x_{t+2\tau}, \dots, x_{t+(m-1)\tau}\}$ with m features can be obtained, where $t=1, 2, \dots, n$ ($= N-(m-1)\tau$), and τ is the lagged time. Based on Eq. (2.4), the ANN forecasting model is formulated as

$$x_{t+(m-1)\tau}^F = f(\mathbf{Y}(t), w, \theta, m, h) = \theta_0 + \sum_{j=1}^h w_j^{out} \varphi(\sum_{i=1}^m w_{ji} x_{t+(i-1)\tau} + \theta_j) \quad (5.1)$$

where $x_{t+(i-1)\tau}$ represents elements in the input vector \mathbf{Y}_t ; ϕ denotes transfer functions; $x_{t+T+(m-1)\tau}^F$ is the single output which stands for the forecasted rainfall at the lead time T . w_{ji} are the weights defining the link between the i th node of the input layer and the j th of the hidden layer; θ_j are biases associated to the j th node of the hidden layer; w_j^{out} are the weights associated to the connection between the j th node of the hidden layer and the node of the output layer; and θ_0 is the bias at the output node. To apply Eq. (5.1) to hydrological predictions, appropriate training algorithm is required for optimizations of w and θ .

5.2 Support Vector Regression

Using the same univariate time series as that in the above ANN model, derived from Eq. (2.4) and Eq. (2.7), the SVR forecast model is given by

$$x_{t+T+(m-1)\tau}^F = f(\mathbf{Y}(t), \omega) = \omega \cdot \phi(\mathbf{Y}(t)) + b \quad (5.2)$$

where the input data $\mathbf{Y}(t)$ in the input space is mapped to a high dimensional feature space via a nonlinear mapping function $\phi(\mathbf{Y}(t))$. The objective of the SVR is to find optimal ω, b and some parameters in kernel function $\phi(\mathbf{Y}(t))$ so as to construct an approximation function of the $f(\bullet)$.

When introducing Vapnik's ϵ -insensitivity error (or loss function), the loss function $L_\epsilon(y, f(\mathbf{Y}(t), \omega))$ on the underlying function can be defined as

$$L_\epsilon(y, f(\mathbf{Y}(t), \omega)) = |y - f(\mathbf{Y}(t), \omega)|_\epsilon = \begin{cases} 0 & \text{if } |y - (\omega \cdot \phi(\mathbf{Y}(t)) + b)| \leq \epsilon \\ |y - (\omega \cdot \phi(\mathbf{Y}(t)) + b)| - \epsilon & \text{otherwise} \end{cases} \quad (5.3)$$

where y represents observed value. Similar to linear SVR (Kecman, 2001; Yu et al., 2006), the nonlinear SVR problem can be expressed as the following optimization problem

$$\begin{aligned}
& \text{minimize } R_{w, \xi_i, \xi_i^*} = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\
& \text{subject to } \begin{cases} y_i - f(\phi(Y_i), w) - b \leq \varepsilon + \xi_i \\ f(\phi(Y_i), w) + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (5.4)
\end{aligned}$$

where Y_i represents $Y(i)$ for simplicity, the term of $\frac{1}{2} \|w\|^2$ reflects generalization,

and the term of $C \sum_{i=1}^n (\xi_i + \xi_i^*)$ stands for empirical risk. The objective in Eq. (5.4) is to minimize them concurrently, which implements SVR to avoid underfitting and overfitting the training data. ξ_i and ξ_i^* are slack variables for measurements “above” and “below” an ε tube (see Figure 2.4). Both slack variables are positive values. C is a positive constant that determines the degree of penalized loss when a training error occurs.

By introducing a dual set of Lagrange multipliers, α_i and α_i^* , the minimization problem can be solved in a dual space. The objective function in dual form can be represented as (Gunn, 1998)

$$\begin{aligned}
& \text{maximize } L_d(\alpha, \alpha^*) = -\varepsilon \sum_{i=1}^n (\alpha_i^* + \alpha_i) + \sum_{i=1}^n (\alpha_i^* - \alpha_i) y_i - \frac{1}{2} \sum_{i,j=1}^n (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) (\phi(Y_i) \cdot \phi(Y_j)) \\
& \text{subject to } \begin{cases} \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \\ 0 \leq \alpha_i^* \leq C, \quad i = 1, \dots, n \\ 0 \leq \alpha_i \leq C, \quad i = 1, \dots, n \end{cases} \quad (5.5)
\end{aligned}$$

By using a “kernel” function $K(Y_i, Y_j) = (\phi(Y_i) \cdot \phi(Y_j))$ to yield inner products in feature space, the computation in input space can be performed. In the present study, Gaussian radial basis function (RBF) was adopted in the form of $K(Y_i, Y_j) = \exp(-\|Y_i - Y_j\|^2 / 2\sigma^2)$. Once parameters α_i , α_i^* , and b_0 are obtained, the final approximation function of the $f(\bullet)$ becomes

$$f(Y_i) = \sum_{k=1}^n (\alpha_k - \alpha_k^*) K(Y_k \cdot Y_i) + b_0, \quad k = 1, \dots, s \quad (5.6)$$

where Y_k stands for the support vector, α_k and α_k^* are parameters associated with

support vector \mathbf{Y}_k , n and s represent the number of training samples and support vectors, respectively. Three parameters (C, ε, σ), however, have to be optimized so that Eq. (5.6) could be used to perform forecasting.

5.3 Modular Models

To construct a modular or hybrid model (for convenience of presentation, the terms of “modular” in the following also includes “hybrid”), the training data have to be divided into several clusters according to different cluster analysis techniques, and then each single model is applied to each cluster. The fuzzy c-means (FCM) clustering technique is adopted in the present study. It is able to generate either soft or crisp clusters. Based on different partition methods on validation data (equal to “testing data” in the present study), the forecasting from the modular model can be conducted in two ways: soft and hard. Soft forecasting means that the validation data can belong to each cluster with different weights. As a consequence, the modular model output would be a weighted average of the outputs of several single models fitted for each cluster of training data. Hard forecasting is that the modular model output is directly from the output of only triggered local model. As mentioned in Chapter 2 that SVR (or similar techniques) has a poor extrapolation, hard forecasting is, therefore, considered in this study.

5.3.1 FCM Clustering

The FCM clustering is based on the theory of fuzzy sets, which was proposed by Bezdek (1981) as an improvement over the hard k-means clustering algorithm. A fuzzy set consists of objects and their respective grades of membership in the set. The grade of membership of an object in the fuzzy set is given by a subjectively defined fuzzy membership function. The value of the grade of membership of an object can range from 0 to 1. The most important feature of fuzzy set theory is the ability to express in numerical format the impression that stems from a grouping of elements into classes that do not have sharply defined boundaries.

Given a set of vectors $\mathbf{Y}_j, j = 1, \dots, n$, each is described by m features, the FCM clustering partitions the set into c clusters, ν_1, \dots, ν_c , and each data point belongs to a cluster to a degree specified by a membership grade u_{ij} , between 0 and 1. One can define a matrix \mathbf{U} comprising the elements u_{ij} , and assume that the summation of degrees of belonging for a data point is equal to 1, i.e., $\sum_{i=1}^c u_{ij} = 1, \forall j = 1, \dots, n$. The goal of the FCM algorithm is to find c cluster centers so that the cost function of dissimilarity measure is minimized. The cost function can be defined by

$$J(\mathbf{U}, \nu_1, \dots, \nu_c) = \sum_{i=1}^c J_i = \sum_{i=1}^c \sum_{j=1}^N u_{ij} d_{ij}^2 \quad (5.7)$$

where ν_i is the cluster center of the fuzzy subset i , and $d_{ij} = \|\nu_i - \mathbf{Y}_j\|$ is the Euclidean distance between the i th cluster center and j th data point. The necessary conditions for Eq. (5.7) to reach its minimum are

$$\nu_i = \frac{\sum_{j=1}^N u_{ij}^\beta \mathbf{Y}_j}{\sum_{j=1}^N u_{ij}^\beta} \quad (5.8)$$

and

$$u_{ij} = \left[\sum_{k=1}^c \left(\frac{d_{kj}}{d_{ik}} \right)^{2/(\beta-1)} \right]^{-1} \quad (5.9)$$

where $\beta \geq 1$ is a tuning parameter which controls the degree of fuzziness in the clustering process, provided that $d_{kj} \neq 0$ for all $0 \leq i \leq c$. If $d_{kj} = 0$, u_{ij} is set the value of 1.

The FCM clustering algorithm is an iterative procedure that satisfies Eq. (5.8) and Eq. (5.9) to minimize Eq. (5.7). The procedure of implementation is shown as follows (Wang et al., 2006b).

- (1) Initialize the membership matrix \mathbf{U} with random values between 0 and 1;
- (2) Calculate c fuzzy clustering centers, ν_1, \dots, ν_c , using Eq. (5.8);
- (3) Compute the cost function according to Eq. (5.7). Stop if either it is below a tolerance value or its improvement over the previous iteration is below a certain threshold;

Compute a new \mathbf{U} using Eq. (5.9). Return to step (2).

The number c of clusters can be different, which is strongly influenced by the studied case. Figure 5.1 illustrates a one-year rainfall series and the FCM clustering results of the reconstructed state space of the rainfall series, where c was taken the value of 2. It can be observed that cluster1 and cluster2 roughly reflect low and high magnitudes of rainfall observations.

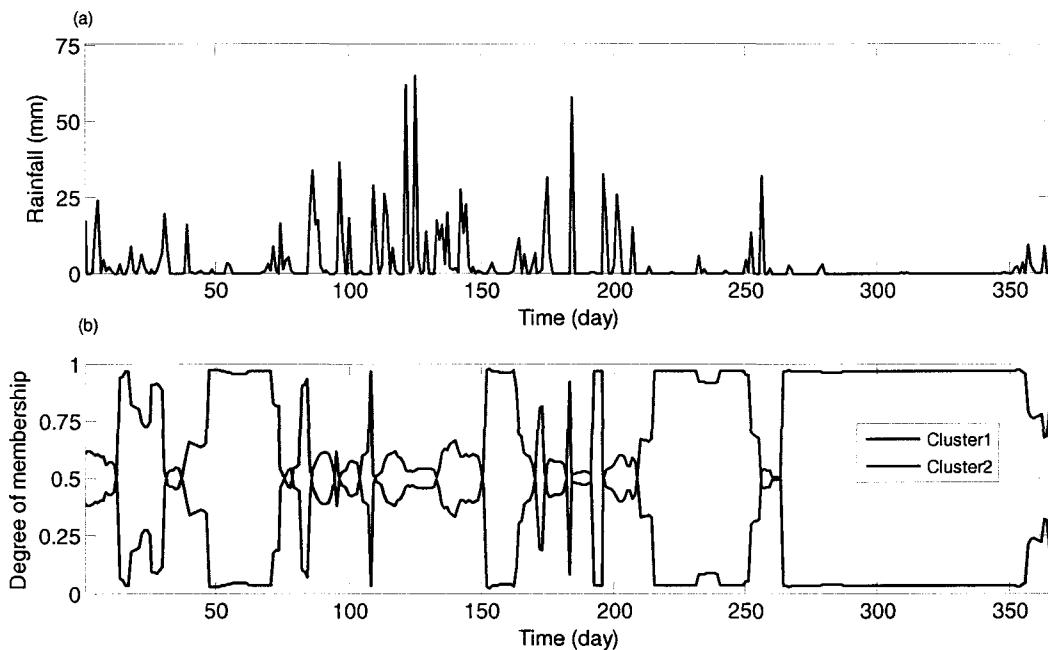


Figure 5.1 Degree of membership of daily rainfall in a representative year using the FCM of two clusters (a) original rainfall series and (b) membership grade

5.3.2 Modular Models

Figure 5.2 displays the schematic diagram of a typical modular model where the training data is partitioned into three clusters. The modular model can be identified as two modes: without data preprocessing (hereafter referred to as “normal mode”) and with data preprocessing, depending on whether or not the data preprocessing operation (the dashed box in Figure 5.2) is implemented. It should be noticed that the parallelity of “determination of model inputs” and “data preprocessing operation” does not mean actions on them concurrently. As a matter of fact, their orders are changeable for different data preprocessing approaches. According to this flow chart,

once input-output pairs are obtained, they are first split into three subsets by the FCM technique, and then each subset is approximated by a single model. The final output of the modular model results directly from one of three local models. When ANN and/or SVR replace these sub-models, MANN, MSVR, and ANN-SVR can be generated.

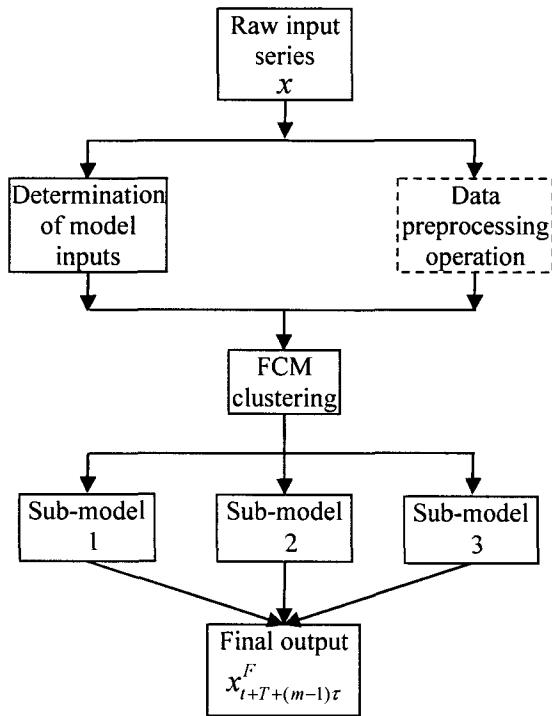


Figure 5.2 Flow chart of hard forecasting using a modular model

5.4 Models Coupled with Data Preprocessing

Four data preprocessing techniques, MA, PCA, SSA, and WA, will be examined for their abilities to improve the performances of data-driven models. The way of the integration of each data preprocessing method with forecast models, is different. The combination procedure is briefly presented as follows.

(1) MA

The combination of studied models with MA is very simple where a new series is obtained by the moving average over the raw data series and then the new series was

used to construct the model inputs.

(2) PCA

PCA aims at reducing dimensionality or preventing collinearity of input variables. Therefore, model inputs are first of all identified, and then PCA is applied to the input variables to yield new model inputs.

(3) SSA and WA

Both SSA and WA are used as filtering tools. The model inputs should be determined first with the development of an original forecasting model, such as ANN. The raw data is then decomposed by SSA or WA into components, and the raw data can be further filtered by selecting p ($\leq L$) appropriate components for reconstruction of the raw data. The optimal components are generally identified by trial and error using a forecast model where the best performance of model is associated with the combination of optimal components.

Part 3

Applications

6 Rainfall Forecasting

Starting from this chapter, several hydrological predictions including rainfall, streamflow, and uncertainty will be carried out using data-driven models coupled with various data preprocessing techniques. In the present chapter, investigation will be made on rainfall forecasts under both monthly and daily scenarios:

- (1) To examine seven model input methods;
- (2) To compare four data-driven models including LR, K-NN, ANN and MANN;
- (3) To investigate the capabilities of three data preprocessing methods of MA, PCA, and SSA.

6.1 Introduction

An accurate and timely rainfall forecast is crucial for reservoir operation and flooding prevention because it can provide an extension of lead-time of the streamflow forecast, larger than the response time of the watershed, in particular for small and medium-sized mountainous basins.

Many studies have been conducted for the quantitative rainfall forecast using diverse techniques including numerical weather prediction models and remote sensing observations (Yates et al., 2000; Ganguly and Bras, 2003; Sheng, et al., 2006; Davolio et al., 2008; Diomede et al. 2008), statistical models (Chu and He, 1995; Chan and Shi, 1999; DelSole and Shukla, 2002; Munot, 2007; Li and Zeng, 2008; Nayagam et al., 2008), chaos theory-based approach (Jayawardena and Lai, 1994), K-NN method (Toth et al, 2000), and soft computing methods including ANN, SVR and FIS (Venkatesan et al., 1997; Silverman and Dracup, 2000; Toth et al., 2000; Pongracz et al., 2001; Sivapragasam et al., 2001; Brath et al., 2002; Chattopadhyay and Chattopadhyay, 2007; Guhathakurta, 2008). Venkatesan et al. (1997) employed the ANN to predict summer monsoon rainfall in India with different meteorological parameters as model inputs. Toth et al. (2000) applied three data-driven models,

ARMA, ANN and K-NN, to short-term rainfall predictions. Results showed that ANN amongst the three models made the best accuracy of runoff forecasting when the predicted rainfalls by the three models were used as inputs of a rainfall-runoff model. FIS was applied to monthly rainfall prediction by Pongracz et al. (2001). Chattopadhyay and Chattopadhyay (2007) constructed an ANN model to predict monsoon rainfall in India depending on the rainfall series by itself.

Recently, the concept of coupling models has attracted more attention in hydrologic predictions. Different coupling methods can broadly be categorized into ensemble models and modular (or hybrid) models. The basic idea behind ensemble models is to build several different or similar models for the same process and to integrate them together (Shamseldin et al., 1997; Shamseldin and O'Connor, 1999; Xiong et al., 2001; Abrahart and See, 2002; Kim et al, 2006). For example, Xiong et al. (2001) used a TSK fuzzy technique to combine several conceptual rainfall-runoff models. Coulibaly et al. (2005) employed an improved weighted-average method to coalesce forecasted daily reservoir inflows from K-NN, conceptual model and ANN. Kim et al. (2006) investigated five coupling methods for improving ensemble streamflow prediction.

Physical processes in rainfall and/or runoff are generally composed of a number of sub-processes. Their accurate modeling by building a single global model is sometimes not possible (Solomatine and Ostfeld, 2008). Modular models are therefore proposed where sub-processes are first of all identified and then separate models (also termed local or expert model) are established for each of them (Solomatine and Ostfeld, 2008). Depending on the soft and hard split of training data, different modular models exist. Soft split means that subset can be overlapped and any overall forecasting output is the weighted-average of each local model (Zhang and Govindaraju, 2000; Shrestha and Solomatine, 2006; Wu et al., 2008). Zhang and Govindaraju (2000) examined the performance of modular networks in predicting monthly discharges based on the Bayesian concept. Wu et al. (2008) employed a distributed SVR for daily river stage prediction. On the contrary, there is no overlapping of subset in hard split and a final forecasted value is explicitly from only one of local models(See and Openshaw, 2000; Hu et al., 2001; Solomatine and Xue,

2004; Sivapragasam and Liong, 2005; Jain and Srinivasulu, 2006; Wang et al., 2006b; Corzo and Solomatine, 2007). Hu et al. (2001) developed a range-dependent network which employed a number of MLPNNs to model the river flow in different flow bands of magnitude (e.g. high, medium and low). Their results indicated that the range-dependent network performed better than the commonly-used global ANN. Solomatine and Xue (2004) used M5 model trees and neural networks in a flood-forecasting problem. Sivapragasam and Liong (2005) divided the flow range into three regions, and employed different SVR models to predict daily flows in high, medium and low regions. Wang et al. (2006b) used a crisp modular ANNs to make soft or crisp predictions for validation data where each local network was trained using the subsets achieved by either a threshold discharge value or a clustering of input spaces.

Apart from adoption of a modular modeling method, the improvement of prediction may be expected by using suitable data preprocessing techniques. Data preprocessing methods from the perspective of signal analysis are crucial because hydrological time series may be viewed as a quasi-periodic signal, which is contaminated by various noises. PCA, WA and SSA were employed in hydrology field by researchers (Sivapragasam et al., 2001; Marques et al., 2006; Hu et al., 2007; Partal and Kişi, 2007; Sivapragasam et al., 2007). Hu et al. (2007) employed PCA as an input data preprocessing tool to improve the prediction accuracy of ANN models for rainfall-runoff transformation. The use of WA to improve rainfall forecasting was conducted by Partal and Kişi (2007). Their results indicated that WA was highly promising. SSA has also been recognized as an efficient data preprocessing technique to avoid the effect of discontinuous or intermittent signals, coupled with neural networks (or similar approaches) for time series forecast (Lisi et al., 1995; Sivapragasam et al., 2001; Baratta et al., 2003). For example, Sivapragasam et al. (2001) proposed a hybrid model of support vector machine (SVM) and SSA for rainfall and runoff predictions. The hybrid model resulted in a considerable improvement in model performance in comparison with the original SVM model. The issue of lagged predictions in ANN was mentioned by some researchers (Dawson and Wilby, 1999; Jian and Srinivasulu, 2004; de Vos and Rientjes, 2005; Muttill and Chau, 2006). de Vos and Rientjes (2005) considered that one of reasons

on lagged predictions was due to the use of previous observed data as ANN's inputs and proposed that an effective solution was to obtain new model inputs by MA over the original data series.

One of the main purposes in this study is to develop a new model of MANN coupled with data preprocessing techniques to improve rainfall forecasting accuracy where seven model input methods and three data preprocessing methods are examined. MANN consists of three local models which are associated with three subsets clustered by the FCM method. To evaluate MANN, LR, K-NN and ANN are used as counterpart models. ANN is first used to find the best model inputs with the help of seven model input methods. Once all forecast models are established, three data-preprocessing methods can be examined. To ensure wider applications of the conclusions, four case studies consisting of two monthly rainfall series and two daily rainfall series from India and China, are explored. The remaining part is structured as follows. Four case studies are described in Section 6.2. Section 6.3 presents modeling methods and their applications to four rainfall series. The optimal model input method and the best data preprocessing technique can be identified. In Section 6.4, main results are shown along with relevant discussions. Section 6.5 presents main conclusions.

6.2 Study Area and Data

Two daily mean rainfall series from Daning and Zhenshui river basins of China, and two monthly mean rainfall series from India and Zhongxian of China, are analyzed in this chapter.

The Daning River, a first-order tributary of the Yangtze River, is located in the northeast of Chongqing city. The daily rainfall data from Jan. 1, 1988 to Dec. 31, 2007 were measured at six raingauges located at the upstream of the study basin (Figure 6.1). The upstream part is controlled by Wuxi hydrology station, with a drainage area of around 2 000 km². The mean areal rainfall series is calculated by Thiessen polygon method (hereafter the averaged rainfall series is referred to as

Wuxi).

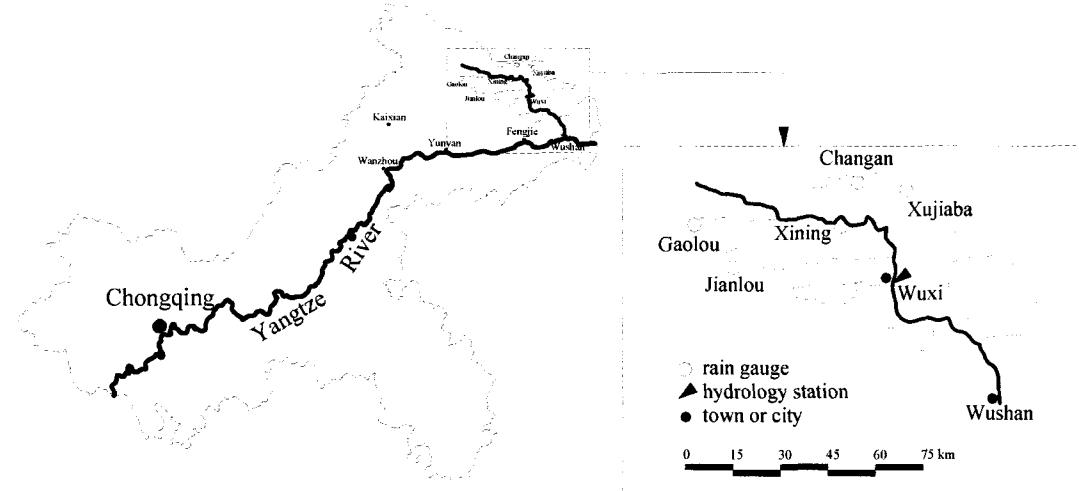


Figure 6.1 Location of Daning river basin (Map of Chongqing in the left, and “Wuxi” watershed in the dashed box)

The Zhenshui basin is located in the north of Guangdong province and adjoined by Hunan province and Jianxi Province. The basin belongs to a second-order tributary of the Pearl River and has an area of $7\ 554\ km^2$. The daily rainfall time series of Zhenwan raingauge was collected between January 1, 1989 and December 31, 1998 (hereafter the averaged rainfall series is referred to as Zhenwan).

The all Indian average monthly rainfall is estimated from area-weighted observations at 306 land stations uniformly distributed over India. The data period spans from January 1871 to December 2007 available at the website <http://www.tropmet.res.in> run by the Indian Institute of Tropical Meteorology.

The other monthly rainfall series is from Zhongxian raingauge which is located in Chongqing city, China. The catchment containing this raingauge belongs to a first-order tributary of the Yangtze River. The monthly rainfall data were collected from January 1956 to December 2007.

Figure 6.2 shows hyetographs of four rainfall series. A linear fit to each hyetograph is denoted by the dashed line. All series appear stationary at least in a weak sense since these linear fits are close to horizontal.

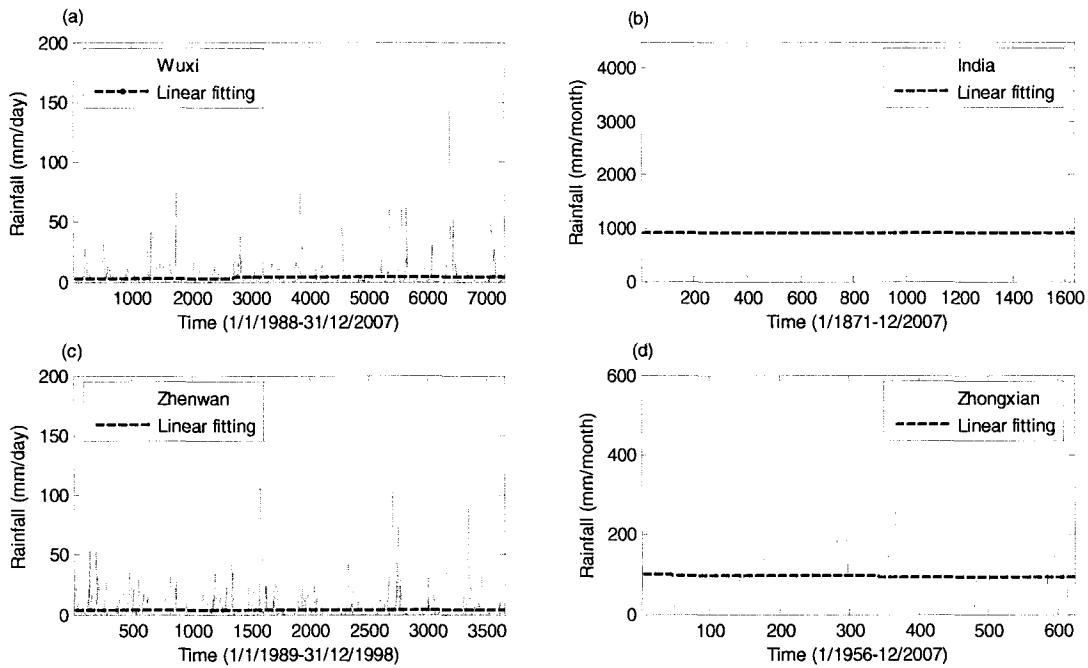


Figure 6.2 Rainfall series of (a) Wuxi, (b) India, (c) Zhenwan, and (d) Zhongxian

In this study, each of data series is partitioned into three parts as training set, cross-validation set and testing set. The training set serves the model training and the testing set is used to evaluate performances of models. The cross-validation set has dual functions: one is to implement an early stopping approach in order to avoid overfitting of the training data and another is to select some best predictions from a large number of ANN's runs. In the present study, 10 best predictions are selected from total 20 ANN's runs. The same data partition is adopted for each rainfall series: the first half of the entire data as training set and the first half of the remaining data as cross-validation set and the other half as testing set.

Table 6.1 presents pertinent information about watersheds and some descriptive statistics of the original data and three data subsets, including mean (μ), standard deviation (S_x), coefficient of variation (C_v), skewness coefficient (C_s), minimum (X_{\min}), and maximum (X_{\max}). As shown in Table 6.1, the training set cannot fully include the cross-validation or testing data. Due to the weak extrapolation ability of ANN, it is suggested that all data be scaled to the interval [-0.9, 0.9] instead of [-1, 1] when ANN employs hyperbolic tangent sigmoid functions as transfer functions in the

hidden layer and output layer.

Table 6.1 Pertinent information for four watersheds and the rainfall data

| Watershed and datasets | Statistical parameters | | | | | | Watershed area and data period |
|------------------------|------------------------|---------------|-------|-------|--------------------|--------------------|--------------------------------|
| | μ (mm) | S_x (mm) | C_v | C_s | X_{\min} (mm) | X_{\max} (mm) | |
| Wuxi | | | | | | | |
| Original data | 3.67 | 10.15 | 0.36 | 5.68 | 0.00 | 154 | Area: |
| Training | 3.81 | 10.94 | 0.35 | 6.27 | 0.00 | 147 | 2 000 km ² |
| Cross-validation | 3.42 | 8.87 | 0.39 | 4.96 | 0.00 | 102 | Data period: |
| Testing | 4.03 | 11.60 | 0.35 | 5.46 | 0.00 | 154 | Jan., 1988- Dec., 2007 |
| Zhenwan | | | | | | | |
| Original data | 4.3 | 11.0 | 0.39 | 4.94 | 0.0 | 159 | Area: |
| Training | 4.3 | 11.2 | 0.38 | 5.60 | 0.0 | 159 | 7554 km ² |
| Cross-validation | 4.7 | 11.2 | 0.42 | 4.22 | 0.0 | 125 | Data period: |
| Testing | 4.0 | 10.9 | 0.37 | 4.97 | 0.0 | 133 | Jan., 1989- Dec., 1998 |
| India | | | | | | | |
| Original data | 906.7 | 951.6 | 1.0 | 0.9 | 3.0 | 3460 | Area: |
| Training | 904.8 | 955.7 | 0.9 | 0.9 | 3.0 | 3393 | all India |
| Cross-validation | 918.2 | 969.5 | 0.9 | 1.0 | 8.0 | 3460 | Data period: |
| Testing | 898.9 | 927.4 | 1.0 | 0.9 | 16.0 | 3232 | Jan., 1871- Dec., 2007 |
| Zhongxian | | | | | | | |
| Original data | 96.2 | 79.2 | 1.2 | 1.2 | 0.0 | 599 | Area: |
| Training | 97.2 | 77.5 | 1.3 | 0.9 | 0.0 | 429 | |
| Cross-validation | 98.6 | 86.8 | 1.1 | 1.9 | 0.0 | 599 | Data period: |
| Testing | 91.8 | 74.9 | 1.2 | 0.8 | 0.0 | 306 | Jan., 1956- Dec., 2007 |

6.3 Applications of Models

For a univariate time series, three commonly used forecasting models have already been described in previous chapters. The LR model is established on the basis of the stepwise linear regression technique (see Chapter 3). The K-NN model can be referred to Eq. (2.2) or Eq. (2.4) since Eq. (2.2) is essentially an approximation to Eq. (2.4) if the lagged time $\tau = 1$ in the reconstruction of the univariate time series. The ANN model can be referred directly to Eq. (5.1). The MANN model consists of three local ANN models, which is described in Figure 5.2.

6.3.1 Implementation framework of rainfall forecasting

Figure 6.3 illustrates the implementation framework of rainfall forecasting where four prediction models can be conducted in two modes: without/with three data preprocessing methods (dashed box). The acronyms in the box “methods for model inputs” represent different methods: LCA (linear correlation analysis), AMI (average mutual information), PMI (partial mutual information), FNN (false nearest neighbors), CI (correlation integral), SLR (stepwise linear regression), and MOGA (ANN based on multi-objective genetic algorithm).

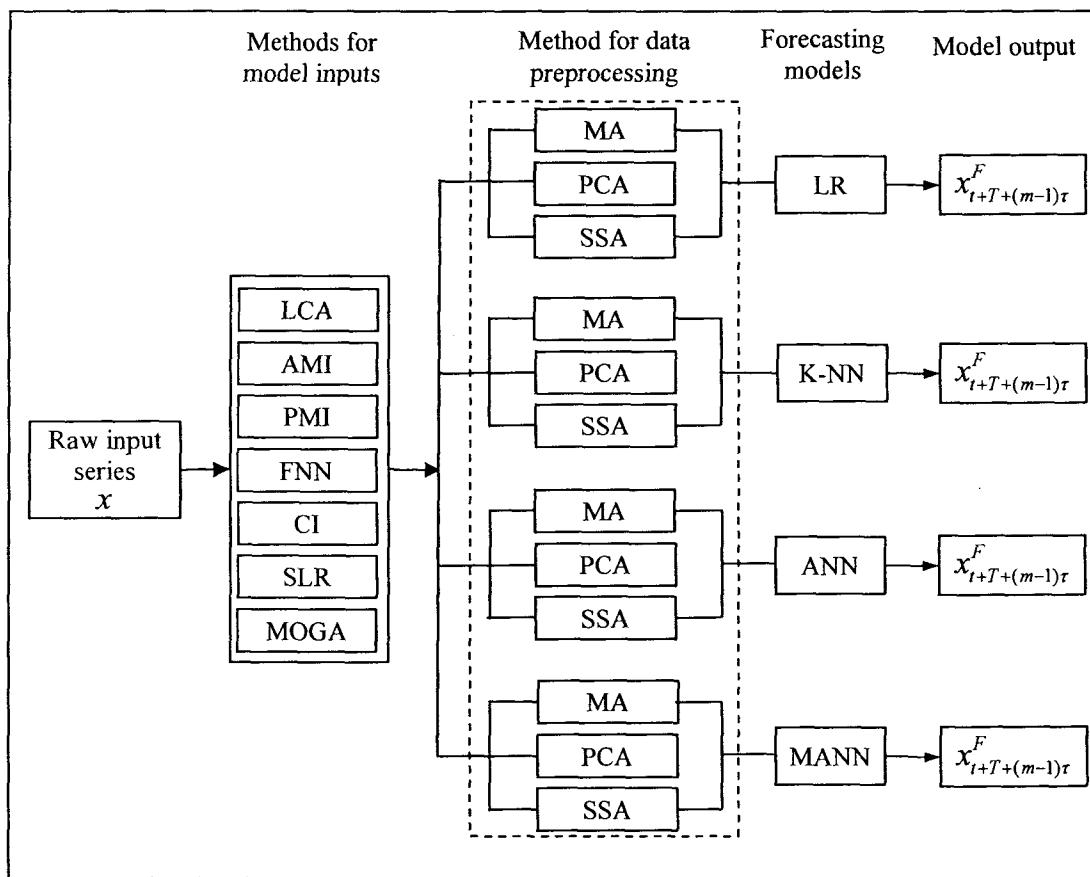


Figure 6.3 Implementation framework of rainfall forecasting models with/without data preprocessing

6.3.2 Evaluation of model performances

The Pearson's correlation coefficient (r) or the coefficient of determination ($R^2 = r^2$),

have been identified as inappropriate measures in hydrologic model evaluation by Legates and McCabe (1999). The coefficient of efficiency (CE) (Nash and Sutcliffe, 1970) is a good alternative to r or R^2 as a “goodness-of-fit” or relative error measure in that it is sensitive to differences in the observed and forecasted means and variances. Legates and McCabe (1999) also suggested that a complete assessment of model performance should include at least one absolute error measure (e.g., RMSE) as necessary supplement to a relative error measure. Besides, the Persistence Index (PI) (Kitanidis And Bras, 1980) is adopted here for the purpose of checking the prediction lag effect. Three measures are therefore used in this study. They are listed below.

$$CE = 1 - \sum_{i=1}^n (y_i - \hat{y}_i)^2 / \sum_{i=1}^n (y_i - \bar{y})^2 \quad (6.1)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (6.2)$$

$$PI = 1 - \sum_{i=1}^n (y_i - \hat{y}_i)^2 / \sum_{i=1}^n (y_i - y_{i-l})^2 \quad (6.3)$$

In these equations, n is the number of observations, \hat{y}_i stands for forecasted rainfall, y_i represents observed rainfall, \bar{y} denotes average observed rainfall, and y_{i-l} is the rainfall estimate from a so-call persistence model (or naïve model) that basically takes the last rainfall observation (at time i minus the lead time l) as a prediction. The value “1” of CE or PI stands for a perfect fit. A small value of PI may imply occurrence of lagged forecasts.

6.3.3 Determination of model inputs

ANN, equipped with the L-M training algorithm and hyperbolic tangent sigmoid transfer functions, is used as a benchmark model to examine seven model input methods in terms of RMSE. Depending on a simplified algorithm for CI from Yu et al. (2000) (downloaded at <http://small.eie.polyu.edu.hk/>), the four rainfall series are identified as non-chaotic since the correlation dimension does not display a characteristic of convergence, in particular, for daily rainfall series (as depicted in Figure 6.4). Results from other six methods are presented in Table 6.2. These results

are based on one step lead prediction and let X_{t+1} be the target value at one-step prediction horizon. It can be seen that there are no significantly differences among these methods in terms of RMSE. In view of the convenience of operation and simplicity of computation, the LCA method is preferred in this study.

Table 6.2 Comparison of methods to determine mode inputs using ANN model

| Watershed | Methods | τ | m | Effective inputs ^a | Identified ANN | RMSE |
|------------------|------------------|--------|-----|--|----------------|--------|
| Wuxi | | | | | | |
| | LCA | 1 | 20 | The last 5 | (5-5-1) | 10.74 |
| | AMI | 1 | 12 | Except for $X_{t-10}, t-9$ | (10-3-1) | 10.91 |
| | PMI ^b | 1 | 12 | $X_{t-1}, t-3, t-5, t-7, t-10, t-4$ | (7-8-1) | 10.85 |
| | FNN | 1 | 20 | The last 14 | (14-3-1) | 11.02 |
| | CI | 4 | 20 | Nil | | |
| | SLR | 1 | 12 | $X_{t-11}, t-7, t-4, t-2, t-1, t$ | (6-3-1) | 10.94 |
| | MOGA | 1 | 12 | $X_t, t-1$ | (2-6-1) | 10.55 |
| Zhenwan | | | | | | |
| | LCA | 1 | 20 | The last 7 | (7-4-1) | 11.03 |
| | AMI | 1 | 12 | Except for $X_{t-11}, t-10, t-9, t-8, t-2$ | (7-5-1) | 10.95 |
| | PMI | 1 | 12 | $X_{t-4}, t-1, t-3, t-11, t-5, t-10, t-6, t-9$ | (10-4-1) | 10.98 |
| | FNN | 1 | 20 | Last 14 | (14-3-1) | 11.08 |
| | CI | 3 | 20 | Nil | | |
| | SLR | 1 | 12 | $X_{t-11}, t-7, t-6, t-3, t-1, t$ | (6-3-1) | 11.01 |
| | MOGA | 1 | 12 | $X_{t-4}, t-7, t-9, t-11$ | (5-8-1) | 10.43 |
| India | | | | | | |
| | LCA | 1 | 20 | the last 12 | (12-5-1) | 256.22 |
| | AMI | 1 | 12 | the last 12 | (12-5-1) | 256.22 |
| | PMI | 1 | 12 | $X_{t-11}, t-10, t-5, t$ | (4-5-1) | 275.06 |
| | FNN | 1 | 20 | the last 5 | (5-9-1) | 286.04 |
| | CI | 4 | 20 | nil | | |
| | SLR | 1 | 12 | except for X_{t-4} | (11-9-1) | 258.13 |
| | MOGA | 1 | 12 | $X_{t-11}, t-9, t-7, t-5, t-4, t-1, t$ | (7-1-1) | 277.57 |
| Zhongxian | | | | | | |
| | LCA | 1 | 20 | the last 13 | (13-3-1) | 51.70 |
| | AMI | 1 | 12 | $X_{t-11}, t-10, t-6, t-5, t-4, t$ | (6-5-1) | 54.67 |
| | PMI | 1 | 12 | $X_{t-11}, t, t-9, t-7, t-7$ | (5-9-1) | 55.39 |
| | FNN | 1 | 20 | the last 4 | (4-7-1) | 59.78 |
| | CI | 3 | 20 | nil | | |
| | SLR | 1 | 12 | $X_{t-11}, t-7, t-6, t-5, t-3, t$ | (6-6-1) | 55.47 |
| | MOGA | 1 | 12 | $X_{t-11}, t-10, t-6, t-3, t$ | (5-2-1) | 53.93 |

Note:^a for the convenience of writing down effective inputs, “ $X_t, t-1$ ” stands for X_t, X_{t-1} ; ^b effective inputs from PMI are in descending order of priority. The original values in the column of m are empirically set.

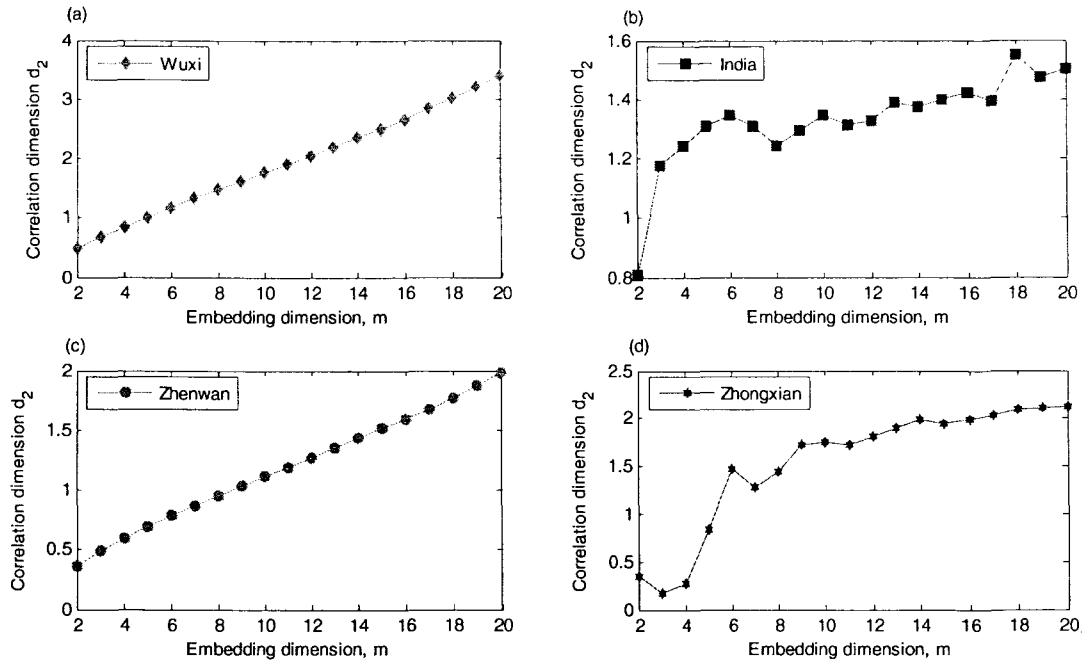


Figure 6.4 Plots of m against d_2 for (a) Wuxi ($\tau = 4$), (b) India ($\tau = 4$), (c) Zhenwan ($\tau = 4$), and (d) Zhongxian ($\tau = 3$).

Figure 6.5 demonstrates the LCA results for Wuxi and Zhenwan. The model inputs are suggested taking previous 5-day rainfalls for Wuxi and previous 7-day rainfalls for Zhenwan because the value of PACF decays within the confidence band around at lag 5 for Wuxi and lag 7 for Zhenwan. Regarding AMI, the effective inputs are also selected based on a 95% confidence limit which is obtained by 200-times bootstraps of the training data. The value of τ for the CI method can be defined when ACF attains the value of zero or below a small value, or AMI reaches the first minimum (Tsonis, 1992). The latter was herein adopted as criterion of the selection of τ . The AMI functions of all four cases are presented in Figure 6.6. Therefore, the values of τ are taken as 4 for Wuxi, 3 for Zhenwan, 4 for India, and 3 for Zhongxian, respectively.

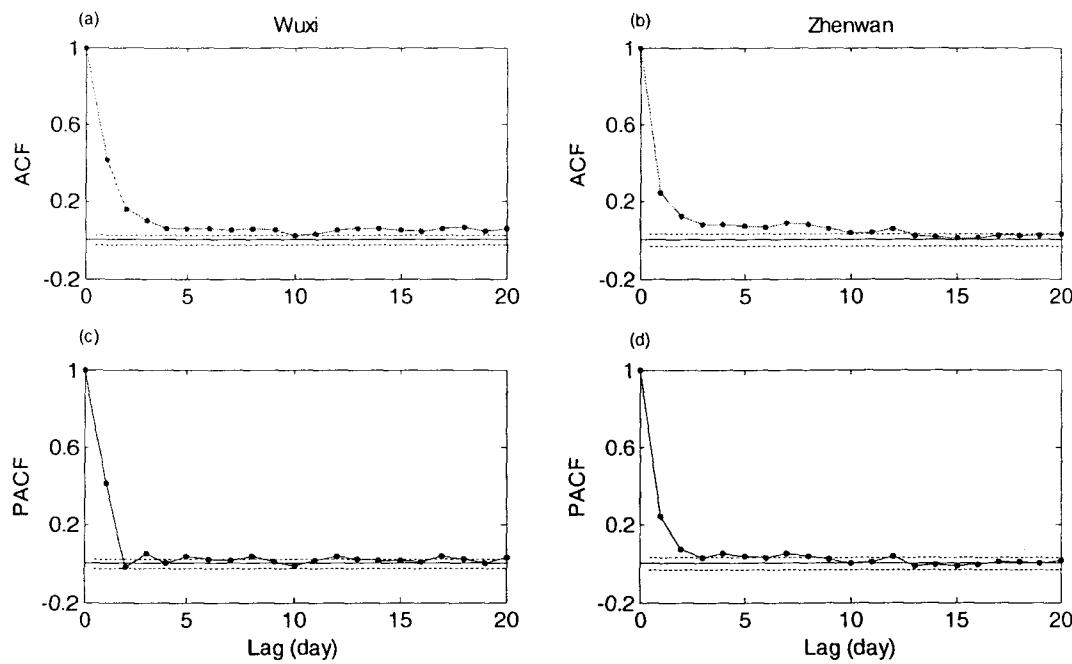


Figure 6.5 Plots of ACF and PACF of the rainfall series with the 95% confidence bounds (the dashed lines), (a) and (c) for Wuxi, and (b) and (d) for Zhenwan

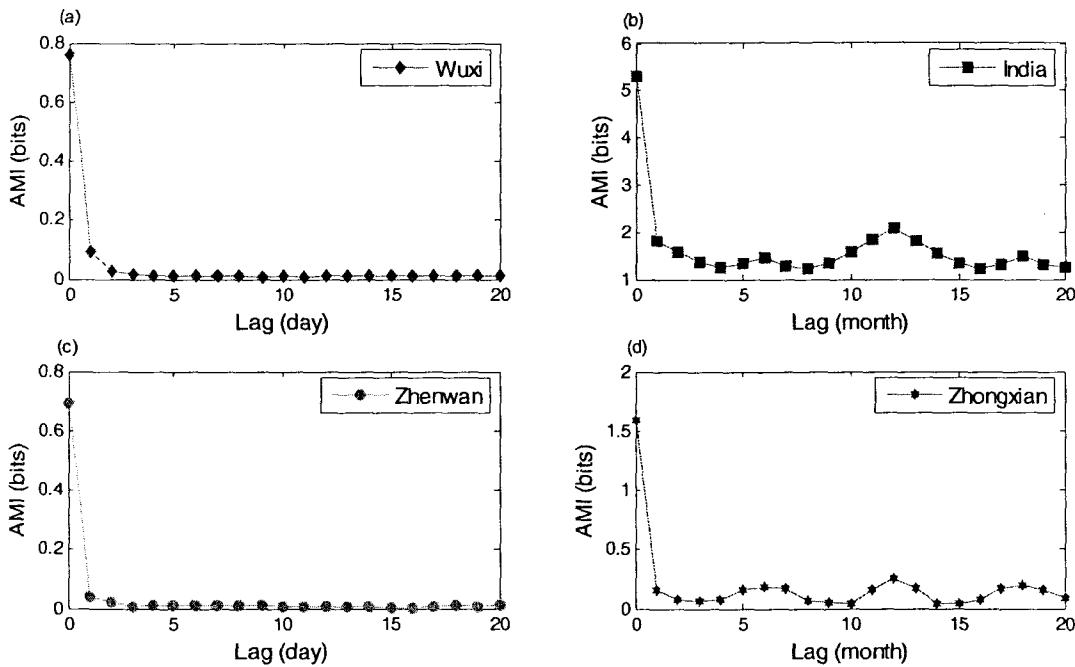


Figure 6.6 Plots of AMI for (a) Wuxi, (b) India, (c) Zhenwan, and (d) Zhongxian.

With respect to the FNN method, using the India series as an example, the sensitivity analysis of the percentage of FNNs (FNNP) on R_{tol} (a threshold value, see Eq.

(3.20)) is demonstrated in Figure 6.7, where τ is set the value of 1 and R_{tol} is from 10 to 30 with a step size of 5. Results show that the FNNP is a little sensitive to R_{tol} and is less than 1% when $R_{tol} \geq 20$ and $m \geq 5$. The current study set R_{tol} as the value of 20. Thus, m is 5 for the India series.

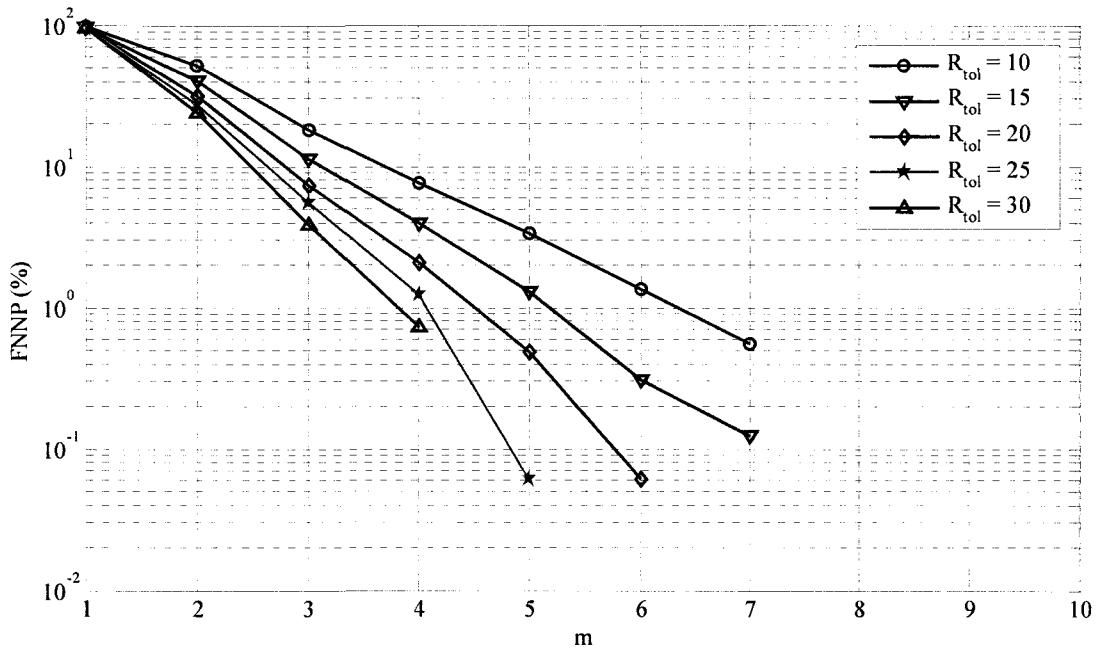


Figure 6.7 Plots of FNNP (in the log scale) with m for India rainfall series with $\tau = 1$ and R_{tol} varying from 10 to 30

6.3.4 Identification of models

The model identification is to determine the structure of a forecasting model by using training data to optimize relevant parameters of model control once model inputs are already obtained. The LR model is built by the SLR technique. In terms of one step prediction (viz., $T = 1$), input variables can be found in Table 6.2. For example, the LR model for Wuxi can be expressed as

$$X'_{t+1} = 0.421X_t - 0.043X_{t-1} + 0.044X_{t-2} + 0.025X_{t-4} + 0.036X_{t-7} + 0.03X_{t-11} \quad (6.1)$$

With respect to K-NN, the model identification consists in finding the optimal K if m -dimensional input vector is determined. Sugihara and Mary (1990) suggested

that the value of K is taken as $K = m+1$. On the other hand, the choice of K should ensure the reliability of the forecasting (Fraser and Swinney, 1986). The check of robustness of $K = m+1$ in terms of RMSE is presented in Figure 6.8, where K is in an interval of [2, 40]. Adopting the value of K as $m+1$ seems reasonable for the current study because the difference between its RMSE and the minimum RMSE is only 2.9% for Wuxi, 2.9% for Zhenwan, 2.6% for India, and 2.0% for Zhongxian, respectively. Consequently, the value of K is taken as 6 for Wuxi ($m=5$), 8 for Zhenwan ($m=7$), 13 for India ($m=12$), and 14 for Zhongxian ($m=13$), respectively.

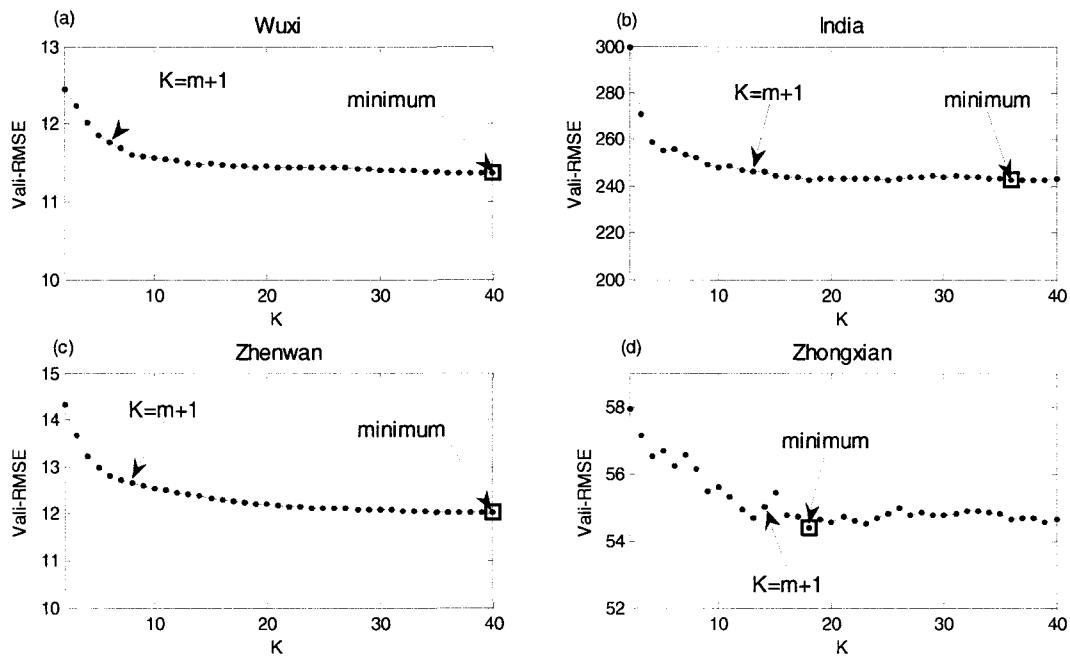


Figure 6.8 Check of robustness of K in KNN for (a) Wuxi, (b) India, (c) Zhenwan, and (d) Zhongxian.

As mentioned in Chapter 2, there are various local forecasting approaches as summarized by Jayawardena and Lai (1994). The equal weight average method was adopted for K-NN. Referring to Eq. (2.2), the formula for one-step lead prediction can be defined as

$$X_{t+1}^F = \frac{1}{K} \sum_{i=1}^K X_{t_i+1} \quad (6.2)$$

where X_{t_i+1} stands for a observed value associated with a neighbor of the current state.

Up to a T -step lead prediction, Eq. (6.2) becomes

$$X_{t+T}^F = \frac{1}{K} \sum_{i=1}^K X_{t_i+T} \quad (6.3)$$

The identification of the structure of ANN is to optimize the number of hidden nodes h in the hidden layer with known model inputs and output. The optimal size h of the hidden layer is found by systematically increasing number of hidden neurons from 1 to 10 until the network performance on the cross-validation set is no longer improved significantly. Based on the LM training algorithm and hyperbolic tangent sigmoid transfer functions, identified configurations of ANN are 5-5-1 for Wuxi, 7-4-1 for Zhenwan, 12-5-1 for India, and 13-3-1 for Zhongxian, respectively. The same method is used to identify the structure of MANN, and the unique difference is that the identification is repeated three times, each time for a local ANN model. Consequently, Structures of MANN are 5-5/7/9-1 for Wuxi, 7-4/8/4-1 for Zhenwan, 12-3/2/5-1 for India, and 13-1/1/1-1 for Zhongxian, respectively.

It is worthwhile to mention that standardization/normalization of the training data is very crucial in improvement of model performance. Two methods can be found in the literature (Dawson and Wilby, 2001; Cannas et al., 2002; Rajurkar et al, 2002; Campolo et al., 2003; Wang et al., 2006b). The standardization (also termed rescaling in some papers) method, as adopted above for model input determination, is to rescale the training data to $[-1, 1]$, $[0, 1]$ or even more narrow interval depending on what kinds of transfer functions are employed in ANN. The normalization method is to rescale the training data to a Gaussian function with a mean of 0 and unit standard deviation, which is by subtracting the mean and dividing by the standard deviation. When the normalization approach is adopted in ANN, We use the linear transfer function (e.g. purelin) instead of the hyperbolic tangent sigmoid function in the output layer. In addition, some studies have indicated that considerations of statistical principles may improve ANN model performance (Cheng and Titterington, 1994; Sarle, 1994). For example, the training data was recommended to be normally distributed (Fortin et al., 1997). Sudheer et al. (2002) suggested that the issue of stationarity should be considered in the ANN development because ANN cannot account for trends and heteroscedasticity in the data. Their results showed that data transformation to reduce the skewness of data was capable of significantly improving the model performance. For the purpose of obtaining better model

performance, four data-transformed schemes are examined:

- Standardizing the raw data (referred to as Std_raw);
- Normalizing the raw data (referred to as Norm_raw);
- Standardizing the n-th root transformed data (referred to as Std_nth_root);
- Normalizing the n-th root transformed data (referred to as Norm_nth_root).

Table 6.3 compares the ANN model performance of the four schemes in terms of RMSE and CE. The Norm_raw scheme is, on the whole, more effective than the Std_raw method. It can also be seen that the effect of the n-th root scheme (taking 3 by trial and error) on the improvement of the performance is basically negligible. Therefore, the Norm_raw scheme is adopted for the later rainfall prediction in the present study.

Table 6.3 Performance comparison of ANN with different data-transformed methods

| Watershed | Data Transformation | RMSE | | | CE | | |
|------------------|---------------------|--------|--------|--------------------|-------|-------|-------|
| | | 1 | 2 | 3 ^a | 1 | 2 | 3 |
| Wuxi | | | | | | | |
| | Std_raw | 10.77 | 11.54 | 11.62 ^b | 0.14 | 0.01 | 0.00 |
| | Norm_raw | 10.57 | 11.49 | 11.59 | 0.17 | 0.02 | 0.00 |
| | Std_nth_root | 11.00 | 12.02 | 12.10 | 0.10 | -0.07 | -0.09 |
| | Norm_nth_root | 11.15 | 12.01 | 12.09 | 0.08 | -0.07 | -0.09 |
| Zhenwan | | | | | | | |
| | Std_raw | 11.03 | 11.11 | 11.16 | 0.03 | 0.02 | 0.01 |
| | Norm_raw | 10.72 | 11.06 | 11.14 | 0.09 | 0.03 | 0.02 |
| | Std_nth_root | 11.25 | 11.68 | 11.75 | -0.01 | -0.09 | -0.10 |
| | Norm_nth_root | 11.34 | 11.70 | 11.74 | -0.02 | -0.09 | -0.09 |
| Wuxi | | | | | | | |
| | Std_raw | 256.22 | 250.51 | 249.46 | 0.92 | 0.93 | 0.93 |
| | Norm_raw | 251.74 | 246.48 | 250.99 | 0.93 | 0.93 | 0.93 |
| | Std_nth_root | 259.81 | 253.42 | 256.43 | 0.92 | 0.93 | 0.92 |
| | Norm_nth_root | 252.75 | 251.95 | 259.00 | 0.93 | 0.93 | 0.92 |
| Zhongxian | | | | | | | |
| | Std_raw | 54.26 | 54.23 | 53.91 | 0.48 | 0.48 | 0.48 |
| | Norm_raw | 52.91 | 53.10 | 52.78 | 0.50 | 0.50 | 0.51 |
| | Std_nth_root | 52.15 | 53.44 | 53.17 | 0.52 | 0.49 | 0.50 |
| | Norm_nth_root | 52.27 | 53.37 | 54.30 | 0.51 | 0.49 | 0.48 |

^a Numbers of “1, 2, and 3” denote one-, two-, and three-day-ahead forecasting;

^b Results are by averaging over 10 best runs from total 20 runs;

6.3.5 Rainfall data preprocessing

(1) MA

The MA operation entails the window length k (see Eq. (4.1)). An appropriate k can be found by systematically increasing k from 1 to 10 to smooth the raw rainfall data. The smoothed data is then used to feed each forecast model. The targeted value of k corresponds to the optimal model performance in terms of RMSE.

(2) PCA

PCA is employed in two ways: one for reduction of dimensionality or preventing collinearity (based on Eq. (4.4)); the other one for noise reduction by choosing leading components (contributing most of the variance of the original rainfall data) to reconstruct rainfall series (based on Eq. (4.9)). The percentage V of total variance (see Eq. (4.8)) is set at three horizons, 85%, 90%, and 95% for principal component selection.

(3) SSA

This approach of filtering a time series to retain desired modes of variability is based on the idea that the predictability of a system can be improved by forecasting the important oscillations in time series taken from the system. The general procedure is to filter the record first and then building of the forecast model is based on the filtered series. To filter the raw rainfall series, the series needs to be decomposed into components with the aid of the SSA. Referring to the SSA theory in Chapter 4, the decomposition by SSA requires identifying the parameter pair (τ, L) (note that the τ can be the same as or different from that employed in the reconstruction of input state space). The choice of L represents a compromise between information content and statistical confidence (Elsner and Tsonis, 1996). The value of an appropriate L should be able to clearly resolve different oscillations hidden in the original signal. However, the present study does not require accurately resolve the raw rainfall signal into trends, oscillations, and noises. A rough resolution can be adequate for the

separation of signals and noises where some leading eigenvalues should be identified.

To select L , a small interval of [3, 10] is examined in the present study. Figure 6.9 shows the relation between singular spectrum (namely, a set of singular values) and singular number L for Wuxi, Zhenwan, India, and Zhongxian. It can be observed that the curve of singular values in each case except for Wuxi tends to level off with the increase of L . Generally, extraction of high-frequency oscillations becomes more difficult with the increase of singular number L (or mode). The criterion of selecting L is empirically defined as: a L is considered as the target only if the singular spectrum under the L can be markedly distinguished. According to this criterion, L is set the value of 7 for India and Zhenwan, 6 for Zhongxian. For Wuxi, all values in the interval satisfy the criterion. To reduce computational demand in later filtering operation L is set a small value of 5 for it. The singular spectrum associated with the selected L is highlighted by the dotted solid line in Figure 6.9.

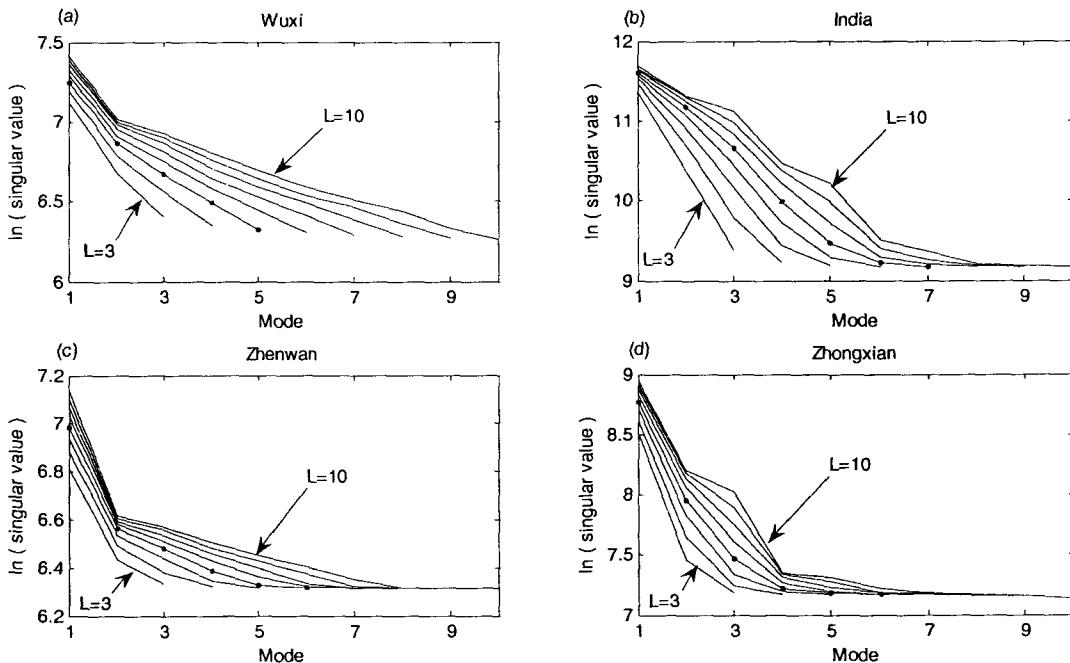


Figure 6.9 Singular Spectrum as a function of lag using various window lengths L for (a) Wuxi, (b) India, (c) Zhenwan, and (d) Zhongxian

Regarding τ , Figure 6.10 presents the results of sensitivity analysis of singular

spectrum on the lag time τ using SSA with the determined L . For daily rainfall series, the singular spectrum can be distinguished only when $\tau=1$. In contrast, the singular spectrum is insensitive to τ in the case of monthly rainfall series. The final parameter pair (τ, L) in SSA are set as (1, 5) for Wuxi, (1, 7) for Zhenwan, (1, 7) for India, (1, 6) for Zhongxian, respectively.

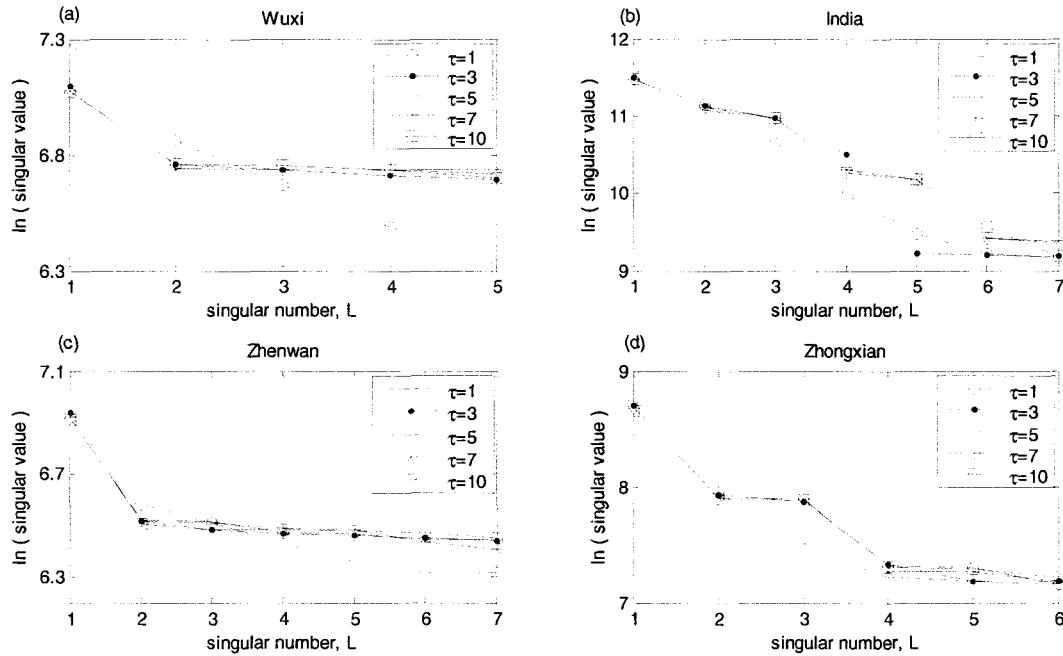


Figure 6.10 Sensitivity analysis of singular Spectrum on varied τ for (a) Wuxi, (b) India, (c) Zhenwan, and (d) Zhongxian

Taking Zhenwan as an example, Figure 6.11 shows the original rainfall series and seven RCs excluding the testing data. Each RC corresponds to a singular value. For instance, RC1 is associated with the largest singular value whereas RC7 responds to the smallest singular value. The RC1 represents an obvious low-frequency oscillation, which demonstrates a similar mode to the original rainfall series. Meantime, the RC7 reflects the highest frequency oscillation. The higher is the frequency of the component, the more possibly it is viewed as a noise.

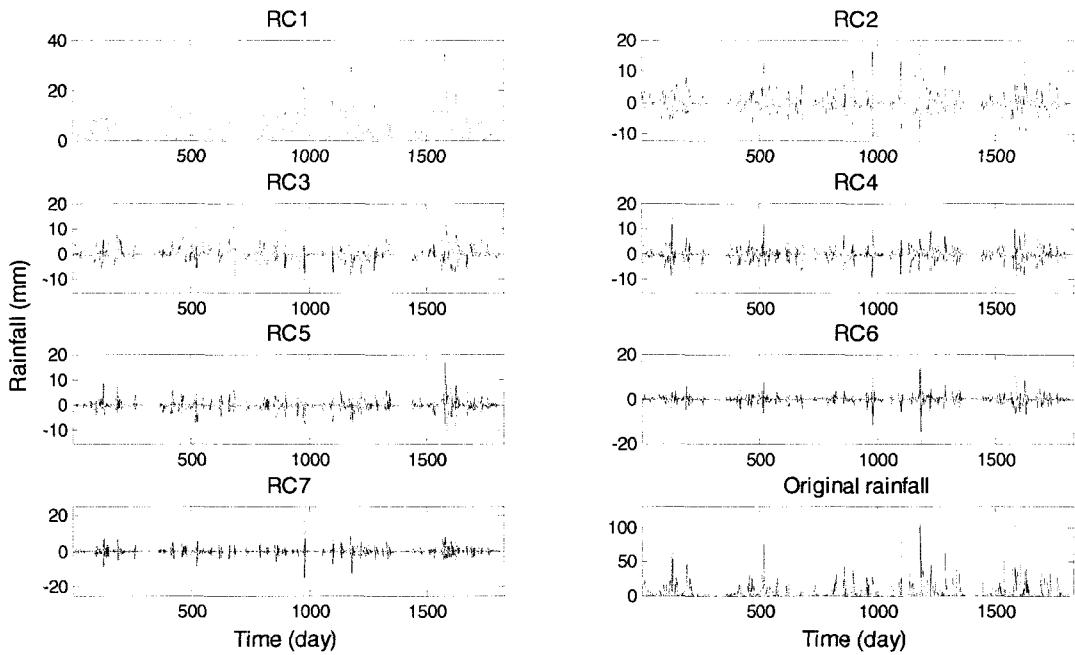


Figure 6.11 Reconstructed components (RCs) and the original series of Zhenwan

6.3.6 Filtering of RCs

Once the rainfall series is decomposed into RCs, the subsequent task is to reconstruct a new rainfall series as model inputs by finding contributing RCs so as to improve predictability of the rainfall series. There is no practical guide on how to identify a contributing or noncontributing component to the improvement of accuracy of prediction. Apparently, a higher-frequency component may be noncontributing. However, the situation may become complicated with the combination of components and change of the prediction horizon. For example, one component viewed as contribution to one-step-ahead prediction may have a negative impact on two-step-lead forecast. Nevertheless, the combined signal of several high-frequency RCs may yield a better input/output mapping than a low-frequency RC. Two filtering methods, supervised and unsupervised, are herein recommended and compared.

(1) Supervised filtering (denoted by SSA1)

Figure 6.12 depicts cross-correlation function (CCF) between RCs and the original rainfall data of Zhenwan. The last plot in this Figure presents the average of CCFs from all seven RCs. The average indicates an overall correlation between input and

output at various lags (also termed prediction horizons). The plot of average CCF shows that the best correlation is positive and occurs at the first lag. RC1 among all 7 RCs exhibits the best positive correlation with the original rainfall series, which is consistent with the revealed fact in Figure 6.11 where RC1 has similar mode to the original Wuxi series. The CCF values for other RCs alternatively change between positive and negative with the increase of the lag. From the perspective of linear correlation, the positive or negative CCF value may indicate that corresponding RC makes a positive or negative contribution to the output of model when the RC is used as the input of model. With this assumption, deleting RCs, which have negative correlations with the model output if the average CCF is positive, may improve the performance of a forecast model. This is the basic idea behind the supervised method.

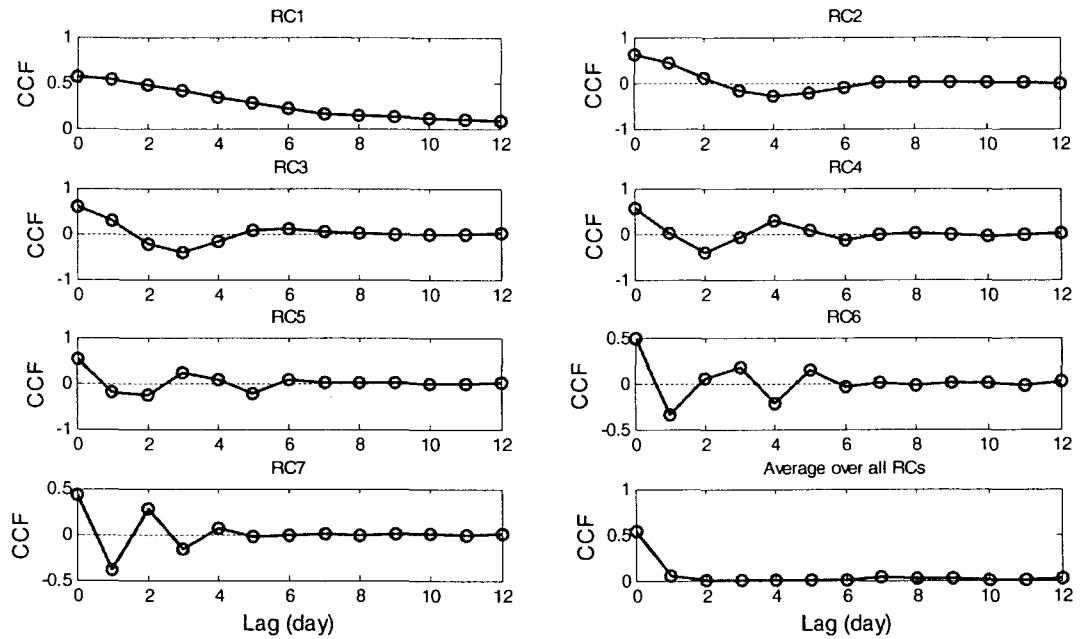


Figure 6.12 Plots of CCF between each RC and the raw rainfall data for Zhenwan

The procedure of the supervised method coupled with ANN is depicted in Figure 6.13. The aim is to find the optimal p ($\leq L$) RCs from all L RCs for each prediction horizon. The procedure can be summarized into three steps: SSA decomposition, correlation coefficients sort, and reconstructed components filter. Operation in each step is bounded by the dashed box. It is worth noting that the filtering method is

based on assumption that combination of components with the same sign in CCF (+ or -) can strengthen the correlation with the model output.

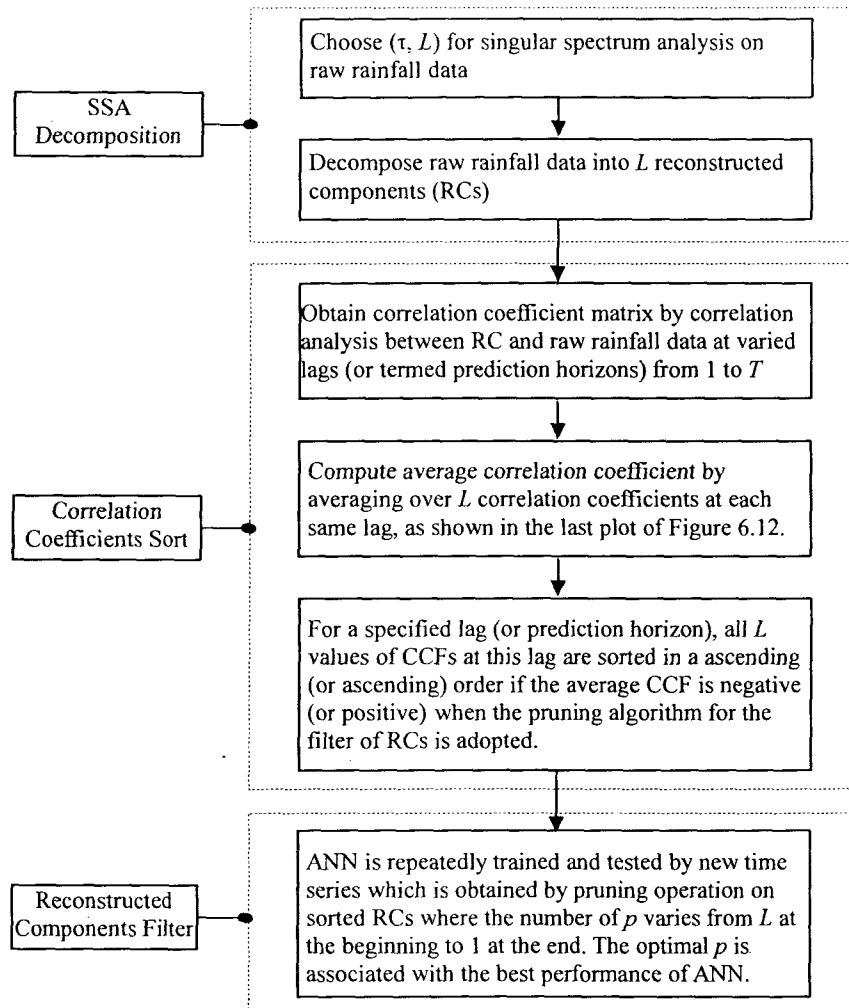


Figure 6.13 Supervised procedure for a forecasting model coupled with SSA

(2)Unsupervised filtering (denoted by SSA2)

There are some drawbacks on the supervised method. The salient one is that the method relies on linear correlation analysis, which disregards the existence of nonlinearity in hydrologic processes. Also, random combinations among all RCs are not taken into account. To overcome these drawbacks, an unsupervised filtering method (also termed enumeration) is recommended in which all input combinations are examined. There are 2^L combinations for L RCs. Therefore, the unsupervised method may be computationally intensive if L is taken a larger number.

6.4 Results and Discussions

This section presents predicted results of various models using two types of modes, namely “normal” mode and “data preprocessing” mode. The “data preprocessing” mode is separately described by MA, PCA, and SSA. To extend one-step-ahead prediction to multi-step-ahead prediction, a direct multi-step prediction method (by directly having the multi-step-ahead prediction as output, also termed static prediction method) is adopted in this study to perform two- and three-step-ahead predictions.

6.4.1 Forecasting with normal mode

Table 6.4 shows the results of three prediction horizons by applying five models including naïve model (see Appendix B) with the normal mode to each case study. The naïve model is used as benchmark in which the forecasted value is directly equal to the last observed value (namely, no change). The naïve model presents the poorest forecasts which may be explained by the fact that it is unlikely to capture any dependence relation. From the perspective of rainfall series, the monthly rainfall can be better predicted than the daily rainfall. Generally, a daily rainfall series, in particular in a semi-humid and semi-dry or dry region, tends to be intermittent and discontinuous due to a large number of no rain periods (dry periods). Two global modeling methods, LR and ANN, mainly capture the zero-zero (or similar extreme low-intensity) rainfall patterns in daily rainfall series because the type of pattern was overwhelmingly dominant in the daily rainfall series. As a consequence, poor performance indices in terms of RMSE, CE, and PI can be observed (depicted in Table 6.4 for Wuxi and Zhenwan). Nevertheless, Table 6.4 also shows that MANN performs the best in each case study. MANN adopted three local ANN models, one for each cluster generated by FCM, which can better capture the mapping relation than using a single global ANN. It can be noticed that MANN is more effective for daily rainfall series than monthly rainfall data, which can be because daily rainfall data is more irregular (or non-periodic) than monthly rainfall series. K-NN for daily rainfall forecasting is even worse than LR although it employs a local prediction approach. Apart from the issue of the selection of K , the performance of K-NN is

also influenced by the similarity of input-output patterns. The smooth monthly rainfall series easily construct similar patterns so that they are well predicted by K-NN. It is worth to note that negative values occasionally appear in the forecasts of ANN or MANN whereas this situation does not happen in the K-NN method.

Table 6.4 Model performances at three forecasting horizons under normal mode

| Watershed | Model | RMSE | | | CE | | | PI | | |
|------------------|-------|-------|--------|--------|-------|-------|-------|------|------|-------|
| | | 1* | 2* | 3* | 1 | 2 | 3 | 1 | 2 | 3 |
| WuXi | | | | | | | | | | |
| | Naïve | 12.2 | 16.0 | 16.5 | 0.05 | -0.61 | -0.72 | 0.00 | 0.00 | 0.00 |
| | LR | 10.9 | 11.9 | 12.0 | 0.12 | -0.05 | -0.07 | 0.28 | 0.41 | 0.43 |
| | K-NN | 11.8 | 12.4 | 12.6 | -0.03 | -0.14 | -0.18 | 0.16 | 0.36 | 0.38 |
| | ANN | 10.6 | 11.5 | 11.6 | 0.17 | 0.02 | 0.00 | 0.32 | 0.45 | 0.47 |
| | MANN | 8.2 | 9.2 | 9.0 | 0.50 | 0.38 | 0.40 | 0.60 | 0.65 | 0.68 |
| Zhenwan | | | | | | | | | | |
| | Naïve | 12.4 | 12.2 | 13.6 | -0.53 | -0.49 | -0.84 | 0.00 | 0.00 | 0.00 |
| | LR | 11.1 | 11.3 | 11.4 | 0.02 | -0.02 | -0.03 | 0.39 | 0.42 | 0.45 |
| | K-NN | 12.7 | 12.7 | 12.8 | -0.27 | -0.28 | -0.30 | 0.21 | 0.28 | 0.31 |
| | ANN | 10.7 | 11.1 | 11.1 | 0.09 | 0.03 | 0.02 | 0.43 | 0.45 | 0.48 |
| | MANN | 7.9 | 9.6 | 9.9 | 0.50 | 0.27 | 0.23 | 0.69 | 0.59 | 0.59 |
| India | | | | | | | | | | |
| | Naïve | 643.1 | 1084.5 | 1399.2 | 0.52 | -0.37 | -1.28 | 0.00 | 0.00 | 0.00 |
| | LR | 286.8 | 301.6 | 302.7 | 0.90 | 0.89 | 0.89 | 0.80 | 0.92 | 0.95 |
| | K-NN | 246.6 | 257.3 | 251.2 | 0.93 | 0.92 | 0.93 | 0.85 | 0.94 | 0.97 |
| | ANN | 245.2 | 245.9 | 247.2 | 0.93 | 0.93 | 0.93 | 0.86 | 0.95 | 0.97 |
| | MANN | 243.3 | 241.8 | 244.4 | 0.93 | 0.93 | 0.93 | 0.86 | 0.95 | 0.97 |
| Zhongxian | | | | | | | | | | |
| | Naïve | 75.7 | 91.9 | 109.2 | -0.03 | -0.51 | -1.13 | 0.00 | 0.00 | -0.02 |
| | LR | 56.1 | 57.7 | 58.4 | 0.44 | 0.41 | 0.39 | 0.46 | 0.60 | 0.71 |
| | K-NN | 55.0 | 56.0 | 57.2 | 0.46 | 0.44 | 0.42 | 0.48 | 0.63 | 0.72 |
| | ANN | 52.5 | 54.4 | 54.3 | 0.51 | 0.48 | 0.48 | 0.52 | 0.65 | 0.75 |
| | MANN | 50.3 | 50.2 | 53.2 | 0.55 | 0.55 | 0.50 | 0.56 | 0.70 | 0.76 |

* Numbers of “1, 2, and 3” denote one-, two-, and three-step-ahead forecasts

Take Wuxi and India data as representative examples, Figure 6.14 shows the scatter plots and hyetographs of the results at one-day-ahead prediction of ANN and MANN using the rainfall data of Wuxi, where the hyetograph is plotted in a selected range for better visual inspection. ANN seriously underestimates a number of moderate- and high-intensity rainfalls. The low values of CE and PI demonstrate that time shift

between the forecasted and observed rainfall may occur, which is further verified by the hyetograph. MANN noticeably improves the accuracy of forecasting in terms of CE and PI. As shown by the scatter plots, the medium-intensity rainfall can be better simulated by MANN although high-intensity rainfalls (or peak value) are still underestimated. Figure 6.15 shows that the scatter plots and hyetographs of the results at one-day-ahead prediction of ANN and MANN using the rainfall data of India. It can be seen from the hyetograph graphs that both ANN and MANN well reproduce corresponding observed rainfall data, which is further revealed by the scatter plots with a low dispersion around the exact fit line.

Figure 6.16 shows the analysis of the lag effect between forecasted and observed rainfall series. The value of CCF at zero lag corresponds to the actual performance (i.e. correlation coefficient) of the model. A target lag is associated with the maximum value of CCF, and is an expression for the mean lag for the forecast. It can be seen from Figure 6.16 that ANN makes fairly obvious lagged predictions for daily rainfall series, and the lag effect can be overcome by MANN. There are 1, 2, and 3 days lag for Wuxi, which are respectively associated with one-, two-, and three-day-ahead forecasting. In contrast, there is no lag effect in monthly rainfall predictions of ANN or MANN.

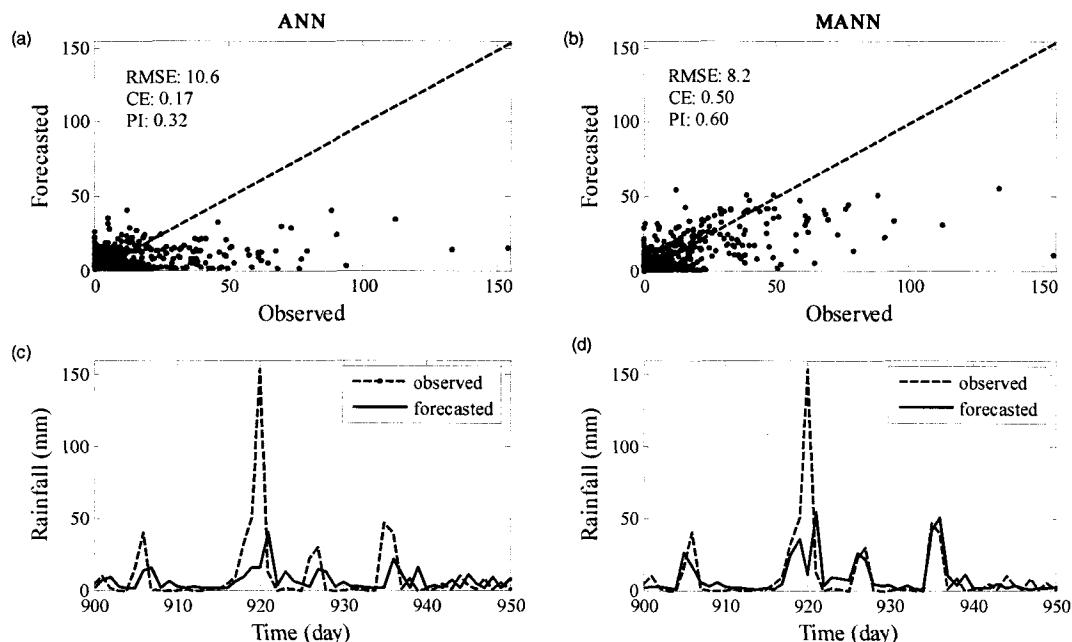


Figure 6.14 Scatter plots and hyetographs of one-step-ahead forecast using ANN and MANN for Wuxi ((a) and (c) from ANN, and (b) and (d) from MANN)

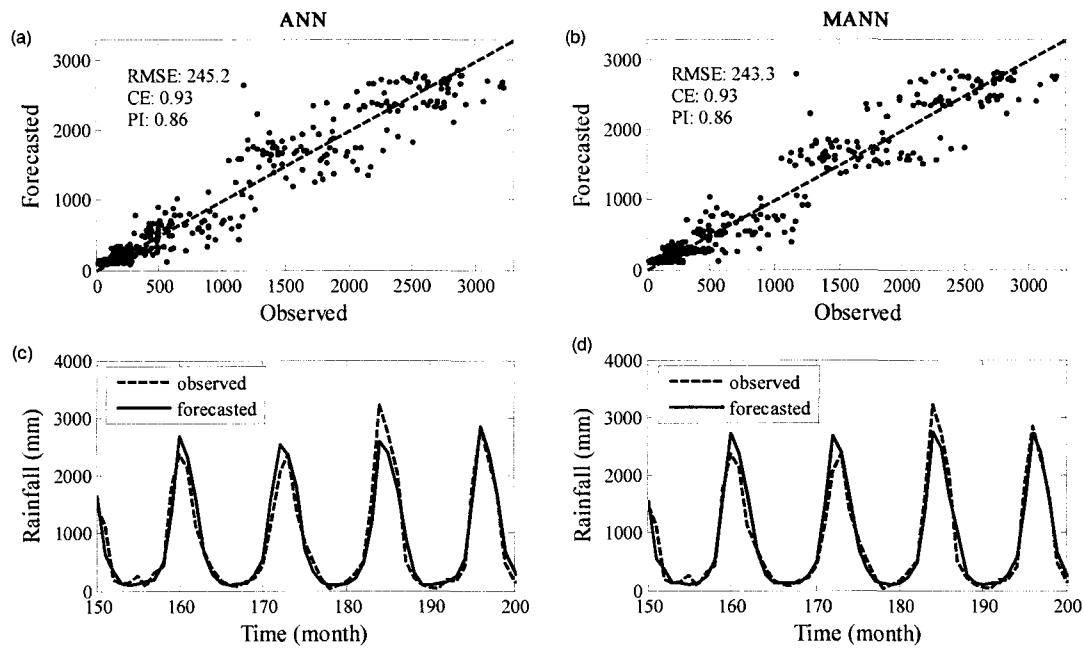


Figure 6.15 Scatter plots and hyetographs of one-step-ahead forecast using ANN and MANN for India ((a) and (c) from ANN, and (b) and (d) from MANN)

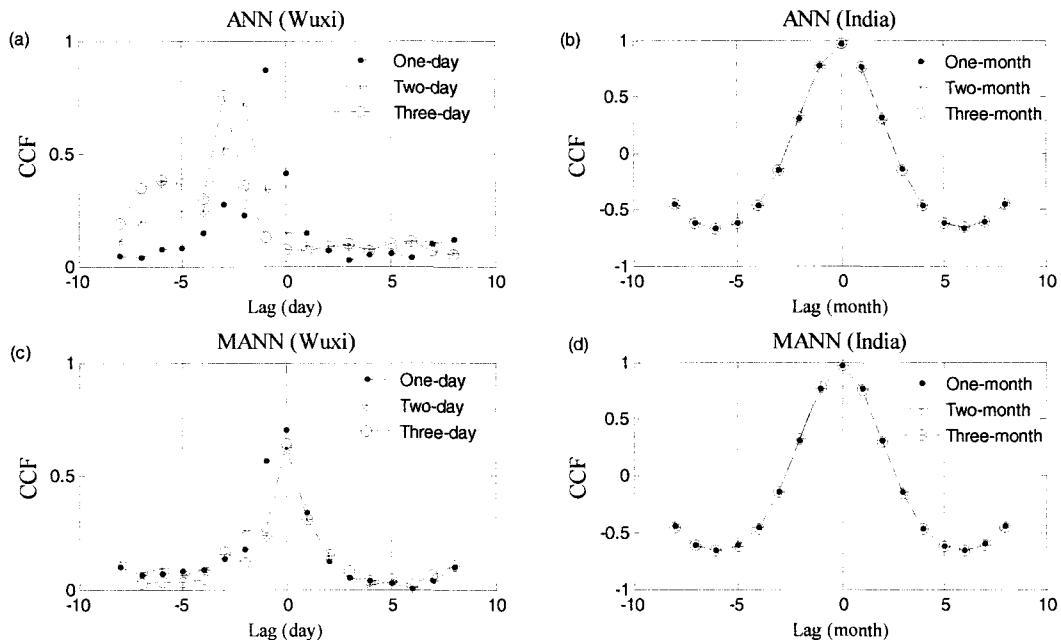


Figure 6.16 CCFs at three forecast horizons for various lags in time of observed and forecasted rainfall of ANN and MANN: Wuxi (left column) and India (right column)

6.4.2 Forecasting with MA

Table 6.5 presents the forecast results of ANN with the “backward” MA (hereafter

referred to as ANN-MA) using the Wuxi rainfall data. The performance indices corresponding to $k=1$ are associated with ANN in the normal mode. The results at each prediction horizon seem to be insensitive to the window length k in view of slight differences among each performance index for k from 1 to 10. Considering the fact that ANNs tend to generate unstable results, the influence of MA on the performance of ANN is negligible. The small values of PI also imply that MA cannot eliminate the lagged forecast from of ANN.

Table 6.5 Model performances of ANN-MA using the Wuxi data

| Prediction horizons | Performance index | Window length (k) for MA | | | | | | | | | |
|---------------------|-------------------|--------------------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| One-step | | | | | | | | | | | |
| | RMSE | 10.60 | 10.70 | 10.63 | 10.72 | 10.77 | 10.70 | 10.83 | 10.83 | 10.72 | 10.68 |
| | CE | 0.17 | 0.15 | 0.16 | 0.15 | 0.14 | 0.15 | 0.13 | 0.13 | 0.15 | 0.15 |
| | PI | 0.32 | 0.31 | 0.32 | 0.31 | 0.30 | 0.31 | 0.29 | 0.29 | 0.31 | 0.31 |
| Two-step | | | | | | | | | | | |
| | RMSE | 11.50 | 11.51 | 11.51 | 11.50 | 11.53 | 11.56 | 11.55 | 11.47 | 11.45 | 11.45 |
| | CE | 0.02 | 0.02 | 0.02 | 0.02 | 0.01 | 0.01 | 0.01 | 0.02 | 0.03 | 0.03 |
| | PI | 0.45 | 0.44 | 0.44 | 0.45 | 0.44 | 0.44 | 0.44 | 0.45 | 0.45 | 0.45 |
| Three-step | | | | | | | | | | | |
| | RMSE | 11.60 | 11.58 | 11.58 | 11.55 | 11.61 | 11.58 | 11.56 | 11.51 | 11.52 | 11.52 |
| | CE | 0.00 | 0.00 | 0.00 | 0.01 | 0.00 | 0.00 | 0.01 | 0.02 | 0.01 | 0.01 |
| | PI | 0.47 | 0.47 | 0.47 | 0.48 | 0.47 | 0.47 | 0.48 | 0.48 | 0.48 | 0.48 |

6.4.3 Forecasting with PCA

As mentioned previously, PCA is used in two ways: one (denoted as PCA1) for reduction of dimensionality (also termed principal component regression, refer to Appendix C.) and the other one (denoted as PCA2) for noise reduction. The results from PCA1 are presented in Table 6.6. The scenario of $V=100\%$ stands for forecasting using models with the normal mode. The results show that PCA1 cannot improve the model performances in terms of RMSE, CE, and PI, which means that the reduction of dimensionality is unnecessary for the present cases. Actually, the

original inputs are characterized by a low dimension. Table 6.7 describes the results from PCA2. According to the results from LR and K-NN (because results from ANN tend to be unstable), a marginal improvement in the model performances can be observed for the Wuxi case whereas the model performances deteriorate for the India case with the decrease of the value of V .

Table 6.6 Multiple-step predictions for Wuxi and India series using LR, K-NN, and ANN with PCA1

| Watershed | Performance index | $V(\%)^1$ | LR | | | K-NN | | | ANN | | |
|--------------|-------------------|-----------|-------|-------|-------|-------|-------|-------|-------|-------|---|
| | | | 1* | 2* | 3* | 1 | 2 | 3 | 1 | 2 | 3 |
| Wuxi | | | | | | | | | | | |
| RMSE | 85 | 11.0 | 11.9 | 12.0 | 12.1 | 12.5 | 12.7 | 10.6 | 11.5 | 11.6 | |
| | 90 | 11.0 | 11.9 | 12.0 | 12.1 | 12.5 | 12.7 | 10.7 | 11.5 | 11.6 | |
| | 95 | 10.9 | 11.9 | 12.0 | 11.8 | 12.4 | 12.6 | 10.6 | 11.5 | 11.6 | |
| | 100 | 10.9 | 11.9 | 12.0 | 11.8 | 12.4 | 12.6 | 10.6 | 11.5 | 11.6 | |
| CE | 85 | 0.12 | -0.05 | -0.07 | -0.04 | -0.15 | -0.19 | 0.16 | 0.01 | 0.00 | |
| | 90 | 0.12 | -0.05 | -0.07 | -0.04 | -0.15 | -0.19 | 0.16 | 0.02 | 0.00 | |
| | 95 | 0.12 | -0.05 | -0.07 | -0.03 | -0.14 | -0.18 | 0.16 | 0.02 | 0.00 | |
| | 100 | 0.12 | -0.05 | -0.07 | -0.03 | -0.14 | -0.18 | 0.17 | 0.02 | 0.00 | |
| PI | 85 | 0.27 | 0.41 | 0.43 | 0.12 | 0.34 | 0.36 | 0.32 | 0.44 | 0.47 | |
| | 90 | 0.27 | 0.41 | 0.43 | 0.12 | 0.34 | 0.36 | 0.31 | 0.45 | 0.47 | |
| | 95 | 0.28 | 0.41 | 0.43 | 0.16 | 0.36 | 0.38 | 0.32 | 0.45 | 0.47 | |
| | 100 | 0.28 | 0.41 | 0.43 | 0.16 | 0.36 | 0.38 | 0.32 | 0.45 | 0.47 | |
| India | | | | | | | | | | | |
| RMSE | 85 | 410.9 | 320.6 | 457.7 | 275.9 | 262.7 | 281.1 | 250.4 | 256.1 | 247.0 | |
| | 90 | 294.9 | 307.8 | 311.1 | 260.3 | 256.2 | 276.8 | 248.1 | 252.3 | 249.0 | |
| | 95 | 291.6 | 305.0 | 304.6 | 255.3 | 256.5 | 265.0 | 247.7 | 254.2 | 251.4 | |
| | 100 | 286.8 | 301.6 | 302.7 | 246.6 | 257.3 | 251.2 | 245.2 | 245.9 | 247.2 | |
| CE | 85 | 0.81 | 0.89 | 0.78 | 0.91 | 0.91 | 0.90 | 0.93 | 0.92 | 0.93 | |
| | 90 | 0.90 | 0.89 | 0.89 | 0.92 | 0.91 | 0.91 | 0.93 | 0.93 | 0.93 | |
| | 95 | 0.90 | 0.89 | 0.89 | 0.92 | 0.91 | 0.91 | 0.93 | 0.92 | 0.93 | |
| | 100 | 0.90 | 0.89 | 0.89 | 0.93 | 0.92 | 0.93 | 0.92 | 0.92 | 0.93 | |
| PI | 85 | 0.61 | 0.92 | 0.90 | 0.81 | 0.93 | 0.96 | 0.85 | 0.94 | 0.97 | |
| | 90 | 0.79 | 0.92 | 0.95 | 0.83 | 0.94 | 0.96 | 0.85 | 0.95 | 0.97 | |
| | 95 | 0.80 | 0.92 | 0.95 | 0.84 | 0.94 | 0.96 | 0.85 | 0.95 | 0.97 | |
| | 100 | 0.80 | 0.92 | 0.95 | 0.85 | 0.94 | 0.97 | 0.84 | 0.94 | 0.97 | |

Note: * “1, 2, and 3” denote one-, two-, and three-step-ahead forecasts; ¹ “ V ” stands for the percentage of total variance.

Table 6.7 Multiple-step predictions for Wuxi and India series using LR, K-NN, and ANN with PCA2

| Watershed | Performance index | $V(\%)^1$ | LR | | | K-NN | | | ANN | | |
|--------------|-------------------|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | | | 1* | 2* | 3* | 1 | 2 | 3 | 1 | 2 | 3 |
| Wuxi | | | | | | | | | | | |
| RMSE | | 85 | 10.8 | 11.6 | 11.6 | 11.5 | 12.2 | 12.7 | 10.6 | 11.5 | 11.6 |
| | | 90 | 10.8 | 11.6 | 11.6 | 11.5 | 12.2 | 12.7 | 10.6 | 11.5 | 11.6 |
| | | 95 | 10.9 | 11.9 | 12.0 | 11.8 | 12.4 | 12.6 | 10.6 | 11.5 | 11.6 |
| | | 100 | 10.9 | 11.9 | 12.0 | 11.8 | 12.4 | 12.6 | 10.6 | 11.5 | 11.6 |
| CE | | 85 | 0.14 | 0.01 | 0.00 | 0.02 | -0.11 | -0.19 | 0.16 | 0.02 | 0.00 |
| | | 90 | 0.14 | 0.01 | 0.00 | 0.02 | -0.11 | -0.19 | 0.16 | 0.02 | 0.00 |
| | | 95 | 0.12 | -0.05 | -0.07 | -0.03 | -0.14 | -0.18 | 0.17 | 0.02 | 0.00 |
| | | 100 | 0.12 | -0.05 | -0.07 | -0.03 | -0.14 | -0.18 | 0.16 | 0.02 | 0.00 |
| PI | | 85 | 0.30 | 0.44 | 0.47 | 0.20 | 0.37 | 0.37 | 0.32 | 0.44 | 0.47 |
| | | 90 | 0.30 | 0.44 | 0.47 | 0.20 | 0.37 | 0.37 | 0.32 | 0.45 | 0.47 |
| | | 95 | 0.28 | 0.41 | 0.43 | 0.16 | 0.36 | 0.38 | 0.32 | 0.45 | 0.47 |
| | | 100 | 0.28 | 0.41 | 0.43 | 0.16 | 0.36 | 0.38 | 0.32 | 0.45 | 0.47 |
| India | | | | | | | | | | | |
| RMSE | | 85 | 482.9 | 413.0 | 800.6 | 254.7 | 248.9 | 253.1 | 241.7 | 247.2 | 249.8 |
| | | 90 | 352.8 | 357.8 | 600.6 | 253.1 | 249.4 | 250.4 | 245.0 | 247.9 | 247.8 |
| | | 95 | 326.5 | 331.9 | 376.1 | 247.8 | 246.1 | 246.1 | 243.5 | 249.0 | 244.8 |
| | | 100 | 286.8 | 301.6 | 302.7 | 246.6 | 257.3 | 251.2 | 247.6 | 252.3 | 247.1 |
| CE | | 85 | 0.73 | 0.80 | -2.74 | 0.92 | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 |
| | | 90 | 0.86 | 0.85 | 0.58 | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 |
| | | 95 | 0.88 | 0.87 | 0.84 | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 |
| | | 100 | 0.90 | 0.89 | 0.89 | 0.93 | 0.92 | 0.93 | 0.93 | 0.93 | 0.93 |
| PI | | 85 | 0.44 | 0.86 | -1.75 | 0.84 | 0.95 | 0.97 | 0.86 | 0.95 | 0.97 |
| | | 90 | 0.70 | 0.89 | 0.81 | 0.85 | 0.95 | 0.97 | 0.86 | 0.95 | 0.97 |
| | | 95 | 0.74 | 0.91 | 0.93 | 0.85 | 0.95 | 0.97 | 0.86 | 0.95 | 0.97 |
| | | 100 | 0.80 | 0.92 | 0.95 | 0.85 | 0.94 | 0.97 | 0.85 | 0.95 | 0.97 |

6.4.4 Forecasting with SSA

Following the procedure in Figure 6.13, the supervised filtering (SSA1) using ANN for RCs of Wuxi and India is illustrated in Figure 6.17. The RMSE associated with the maximum number of p (for instance $p=5$ for Wuxi) represents the performance of ANN with the normal mode. The optimal p corresponds to the minimum RMSE, which can be found by systematically deleting RCs one at a time. Consequently, numbers of chosen optimal p RCs in three forecasting horizons are 3, 2, and 1 for Wuxi, and 1, 3, and 5 for India, respectively. In terms of RMSE, SSA1 substantially improves the performance of ANN. Similar findings are also achieved when SSA1 is coupled with LR or K-NN. A comprehensive description for these

findings is presented in Table 6.8.

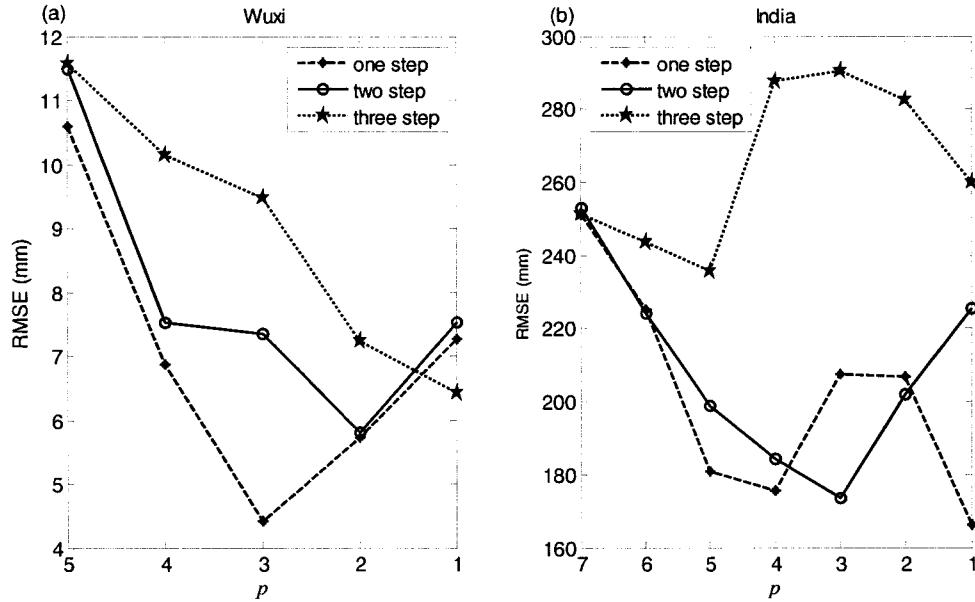


Figure 6.17 Performance of ANN with SSA1 in terms of RMSE as a function of p ($\leq L$) selected RCs at various prediction horizons: (a) Wuxi and (b) India

On the other hand, the unsupervised filtering method (SSA2) is based on enumeration of combinations of all RCs. Selection of the optimal p RCs cannot be presented in a graphical form. Table 6.8 shows selected p at various prediction horizons using LR, K-NN, and ANN in conjunction with SSA1 and SSA2. A large amount of information can be extracted from Table 6.8. First of all, a considerable improvement in the model performance is achieved by each forecast model in conjunction with SSA1 or SSA2, compared with results in Table 6.4. From the perspective of rainfall series, the accuracy of daily rainfall prediction is significantly improved in comparison to that in the normal mode. Secondly, as expected, results from SSA2 are superior to or at least equivalent to those from SSA1 since the former examines each combination of RCs in search of the optimal p . SSA2 is therefore considered as an efficient and effective method if the number L of RCs is small. For the present four cases, the SSA2 method is appropriate due to the small number of RCs. Once L is large, say 40 or 50, SSA1 may be a good alternative where a relative optimal forecast can be guaranteed. Additionally, it should be noticed that the optimal p are different at three forecasting horizons. Finally, Table 6.8 also shows

that ANN among three models performs the best with SSA1 or SSA2, which is consistent with the result in the normal mode.

Table 6.8 Optimal p RCs for model inputs at various forecasting horizons

| Watershed Model | Prediction horizons | Supervised filter (SSA1) | | | Unsupervised filter (SSA2) | | |
|------------------|---------------------|--------------------------|--------|-----------------|----------------------------|------|--|
| | | Optimal p RCs | RMSE | Optimal p RCs | RMSE | | |
| Wuxi | | | | | | | |
| LR | 1 | 1* | 2* | 6.01 | 1 2 | 6.01 | |
| | 2 | 1 5 | 7.73 | 1 5 | 7.73 | | |
| | 3 | 1 | 8.40 | 1 | 8.40 | | |
| K-NN | 1 | 1 | 8.02 | 2 3 | 7.17 | | |
| | 2 | 1 | 8.41 | 2 4 | 8.03 | | |
| | 3 | 1 | 9.99 | 2 | 9.69 | | |
| ANN | 1 | 1 2 3 | 4.43 | 1 2 3 | 4.43 | | |
| | 2 | 1 5 | 5.82 | 1 5 | 5.57 | | |
| | 3 | 1 | 6.42 | 1 | 6.25 | | |
| Zhenwan | | | | | | | |
| LR | 1 | 1 2 3 | 7.19 | 1 2 3 | 7.19 | | |
| | 2 | 1 7 2 | 7.99 | 1 2 7 | 7.99 | | |
| | 3 | 1 5 | 8.81 | 1 5 | 8.81 | | |
| K-NN | 1 | 1 2 3 4 5 | 9.84 | 3 6 | 7.64 | | |
| | 2 | 1 | 9.72 | 3 6 | 8.95 | | |
| | 3 | 1 | 10.33 | 2 5 | 10.24 | | |
| ANN | 1 | 1 2 3 4 | 5.55 | 5 6 7 | 5.02 | | |
| | 2 | 1 7 2 | 5.84 | 3 7 | 5.51 | | |
| | 3 | 1 5 | 6.58 | 3 7 | 5.56 | | |
| India | | | | | | | |
| LR | 1 | 2 1 3 4 | 185.95 | 1 2 3 4 | 185.95 | | |
| | 2 | 1 2 | 237.85 | 1 2 | 237.85 | | |
| | 3 | 3 4 2 7 1 | 299.14 | 1 2 6 | 287.16 | | |
| K-NN | 1 | 2 1 3 4 5 | 236.90 | 1 2 5 6 | 236.39 | | |
| | 2 | 1 2 | 247.44 | 1 2 5 | 242.65 | | |
| | 3 | 3 4 2 7 | 249.43 | 1 2 5 6 | 243.86 | | |
| ANN | 1 | 2 | 166.58 | 1 7 | 164.70 | | |
| | 2 | 1 2 7 | 173.57 | 1 2 7 | 166.30 | | |
| | 3 | 3 4 2 7 1 | 235.59 | 1 5 7 | 172.56 | | |
| Zhongxian | | | | | | | |
| LR | 1 | 1 2 | 40.06 | 1 2 | 40.06 | | |
| | 2 | 1 2 6 | 41.87 | 1 6 | 39.44 | | |
| | 3 | 3 6 2 4 1 | 58.29 | 1 5 | 41.53 | | |
| K-NN | 1 | 1 2 | 51.78 | 1 2 | 51.78 | | |
| | 2 | 1 2 | 53.86 | 1 2 3 | 53.32 | | |
| | 3 | 3 6 2 | 54.15 | 2 3 | 53.16 | | |
| ANN | 1 | 1 2 3 | 38.39 | 1 5 6 | 34.09 | | |
| | 2 | 1 | 44.49 | 1 5 6 | 39.45 | | |
| | 3 | 3 6 | 46.14 | 1 5 | 34.94 | | |

Note: * numbers of “1, 2” stand for RC1 and RC2, and these numbers in the SSA1 column is in a descending order of CCFs shown in Figure 6.12.

In the normal mode, MANN has been proved to be superior to ANN, in particular, for daily rainfall forecasting. As an attempt to improve the accuracy of rainfall

forecasting, MANN is also coupled with SSA2. Table 6.9 demonstrates the results in terms of RMSE, CE, and PI using MANN compared with ANN. Good accuracies of prediction are made by MANN or ANN. It can be seen from the values of PI that the prediction lag effect is completely eliminated. The model performance does not markedly deteriorate with the increase of the forecasting lead. Results also show that MANN still remains a salient superiority over ANN in the SSA2 mode for both daily and month rainfall series.

Table 6.9 Model performances at three forecasting horizons using MANN and ANN with the SSA2

| Watershed | Model | RMSE | | | CE | | | PI | | |
|------------------|-------|-------|-------|-------|------|------|------|------|------|------|
| | | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| WuXi | | | | | | | | | | |
| | ANN | 4.43 | 5.57 | 6.25 | 0.84 | 0.78 | 0.70 | 0.87 | 0.88 | 0.84 |
| | MANN | 3.63 | 4.32 | 3.93 | 0.90 | 0.86 | 0.89 | 0.92 | 0.92 | 0.94 |
| Zhenwan | | | | | | | | | | |
| | ANN | 5.02 | 5.51 | 5.56 | 0.81 | 0.75 | 0.71 | 0.88 | 0.85 | 0.84 |
| | MANN | 3.18 | 3.20 | 3.31 | 0.92 | 0.92 | 0.91 | 0.95 | 0.95 | 0.95 |
| India | | | | | | | | | | |
| | ANN | 164.7 | 166.3 | 172.6 | 0.97 | 0.97 | 0.97 | 0.95 | 0.98 | 0.99 |
| | MANN | 144.2 | 145.1 | 157.4 | 0.98 | 0.98 | 0.97 | 0.95 | 0.98 | 0.99 |
| Zhongxian | | | | | | | | | | |
| | ANN | 34.09 | 39.45 | 34.94 | 0.84 | 0.71 | 0.83 | 0.84 | 0.80 | 0.92 |
| | MANN | 28.58 | 32.24 | 32.69 | 0.86 | 0.82 | 0.82 | 0.86 | 0.88 | 0.91 |

One-step lead estimates of MANN and ANN with the help of SSA2 are shown in Figure 6.18 (Wuxi) and Figure 6.19 (India) in the form of hyetographs and scatter plots (the former is plotted in a selected range for better visual inspection). Compared with Figure 6.14, each scatter plot in Figure 6.18 is closer to the exact line, which means that the daily rainfall process is appropriately fitted. Nevertheless, some peak values still remain mismatched although MANN shows a better ability to capture the peak value than ANN. Regarding the monthly rainfall series, the scatter plots with perfect match of the diagonal indicates that the rainfall process is perfectly reproduced. The representative hyetograph shows that the peak times and peak values are also accurately predicted.

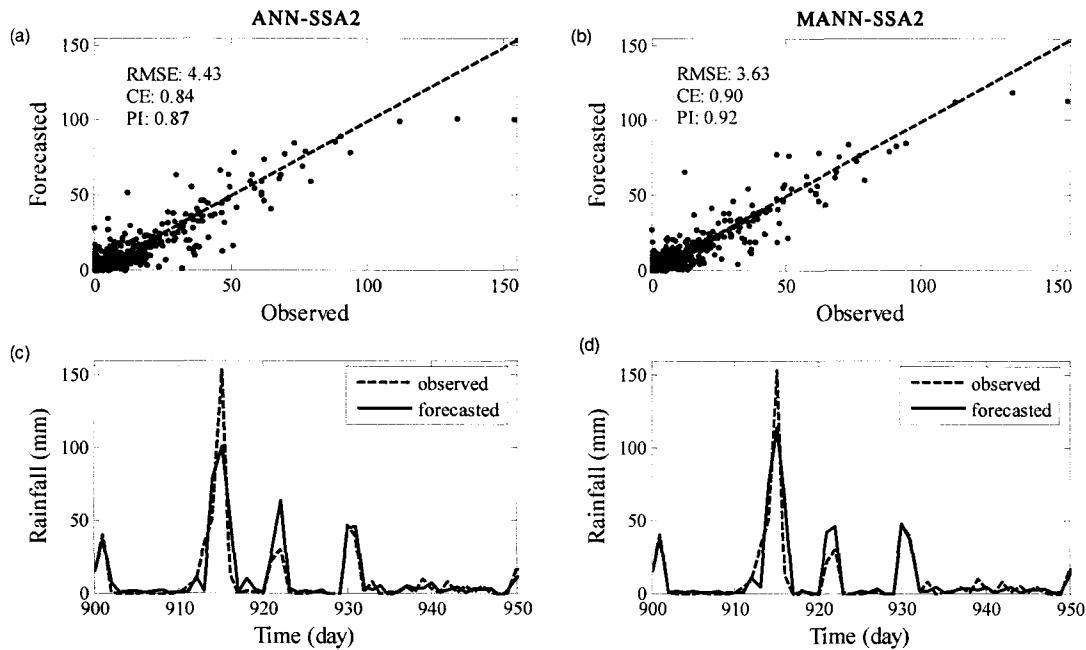


Figure 6.18 Scatter plots and hyetographs of one-step-ahead forecast using ANN and MANN with SSA2 for Wuxi ((a) and (c) from ANN, and (b) and (d) from MANN)

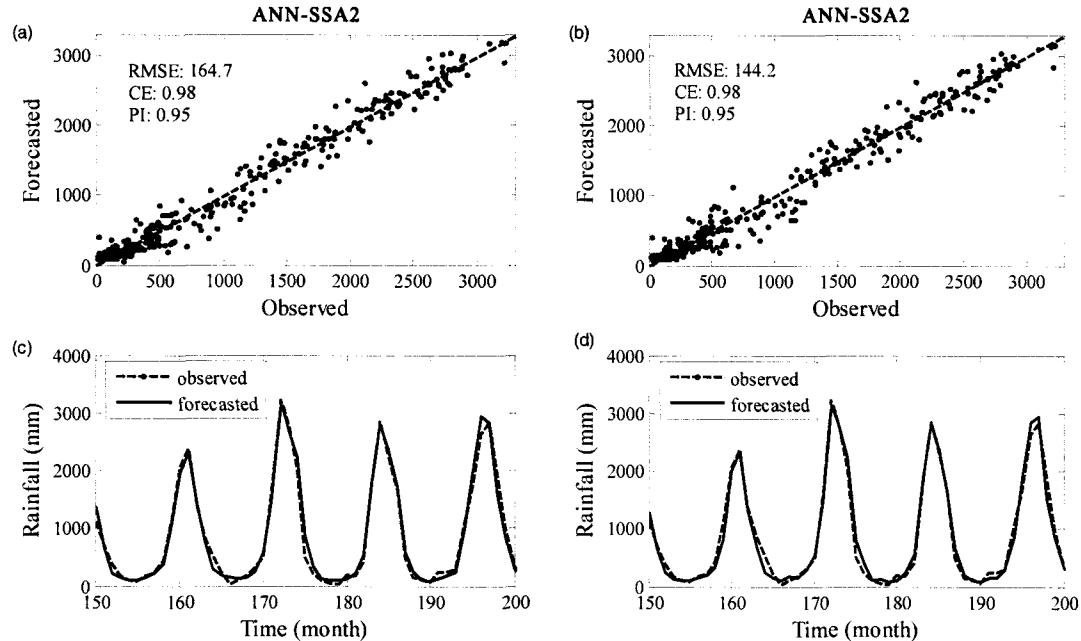


Figure 6.19 Scatter plots and hyetographs of one-step-ahead forecast using ANN and MANN with SSA2 for India ((a) and (c) from ANN, and (b) and (d) from MANN)

Figure 6.20 presents the correlation analysis between observed and forecasted rainfall from ANN and MANN using the Wuxi and India series, respectively. Compared with Figure 6.16, the lagged prediction of ANN is completely eliminated

by SSA2 since the maximum CCF occurs at zero lag. The larger the CCF at zero lag is, the better the model performance is.

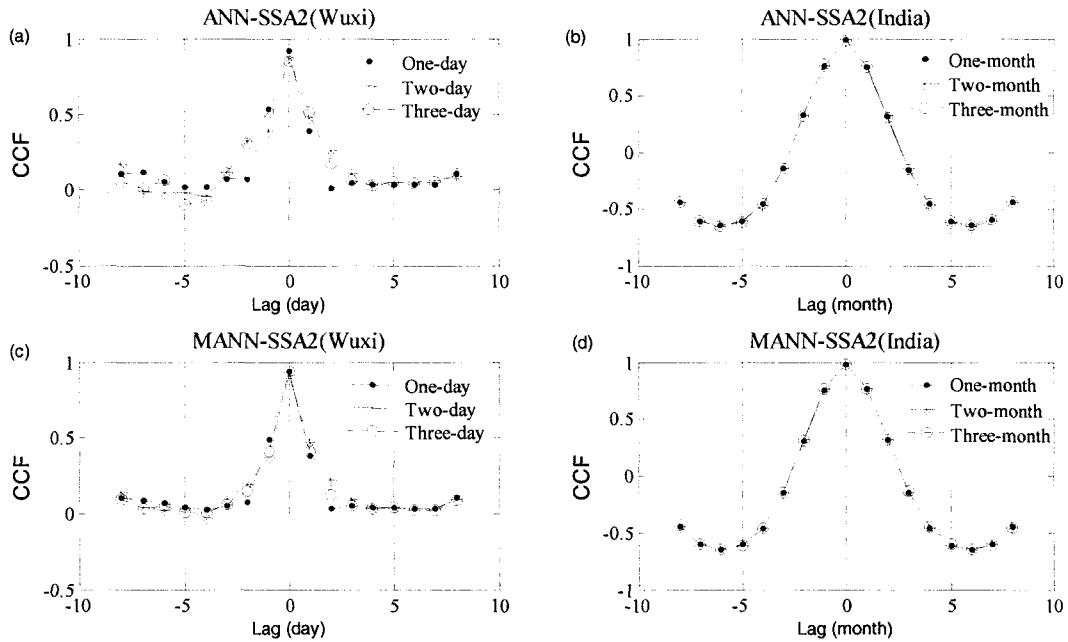


Figure 6.20 CCFs at three forecast horizons using ANN and MANN with Wuxi and India ((a) and (c) for Wuxi and (b) and (d) for India)

A simple method for the check of model performance is to draw a histogram of errors. If the histogram is characterized by a quasi-normal distribution with mean near zero, which indicates that errors basically pass the test for “whiteness”. Figure 6.21 (Wuxi) and Figure 6.22 (India) present error distributions at one-step-ahead forecast using LR, K-NN, ANN, and MANN with the normal mode and SSA2 mode, respectively. With respect to the daily rainfall series in Figure 6.21, all histograms are far from a quasi-normal distribution although the skewness of the histogram in the normal mode is substantially adjusted by SSA2. The long double tails in the histogram is derived from inaccurate forecasts of peak values. Therefore, a further study should be focused on improving prediction of peak values so as to enhance these models' performance. On the contrary, each histogram of errors for the monthly rainfall forecasting using LR, K-NN, ANN, and MANN basically satisfies the normal distribution, which means that these forecasting models are reasonably identified for the India rainfall data. It can be observed that all histograms in the SSA2 mode

except for that from K-NN are remarkably improved in view of the shorter double tails.

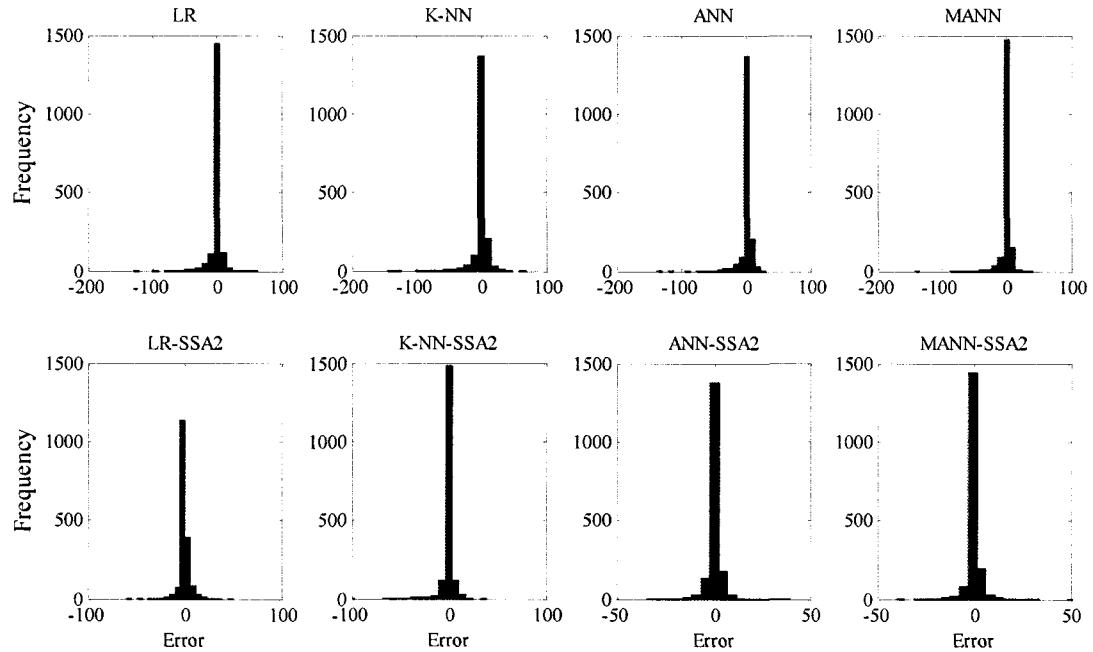


Figure 6.21 Histograms of absolute prediction errors of LR, K-NN, ANN, and MANN with the normal mode and SSA2 mode using the Wuxi data

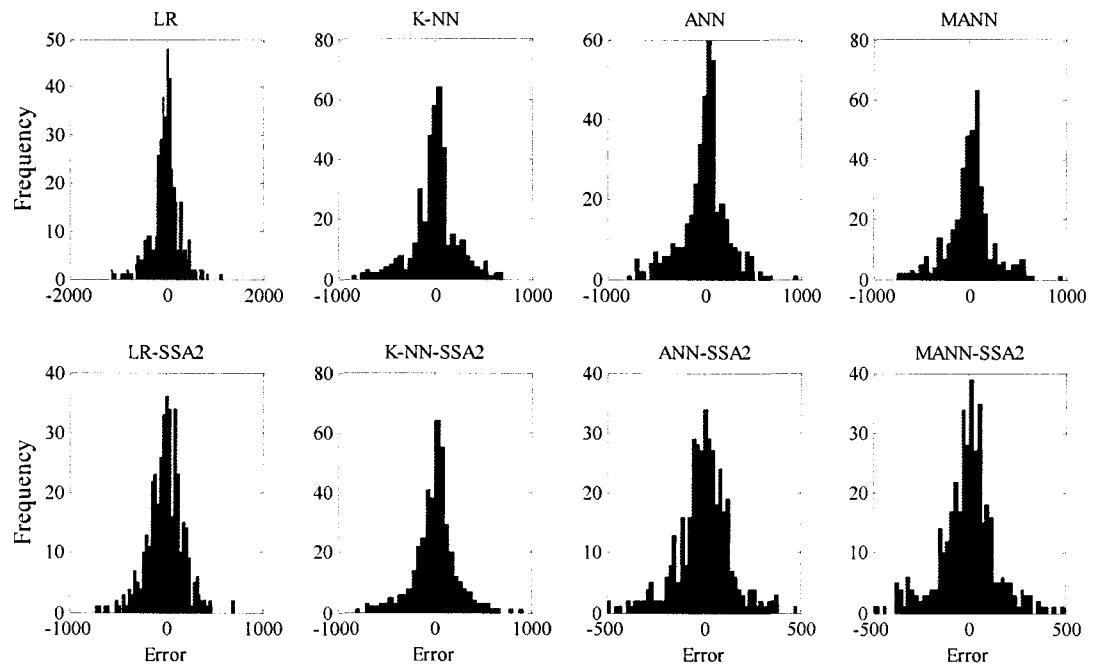


Figure 6.22 Histograms of absolute prediction errors of LR, K-NN, ANN, and MANN with the normal mode and SSA2 mode using the India data

6.4.5 Discussions

The following part concerns some discussions about forecasting models and the effects of the SSA technique.

(1) About the investigation of effects of SSA

Figure 6.23 shows that a large number of zeros and near zeros occur in the original Wuxi rainfall which makes the series discontinue. Using the intermittent series is difficult to reconstruct similar input patterns for a forecast model. Thus, depending on those reconstructed input patterns, these data-driven models based on pattern training, for example, ANNs, tend to be unfeasible. In contrast, the rainfall series preprocessed by SSA becomes smoother where most of zeros are replaced by nonzero values. New input vectors from the reconstructed rainfall series are characterized by better repetition of patterns so that they are reproduced easier by ANN.

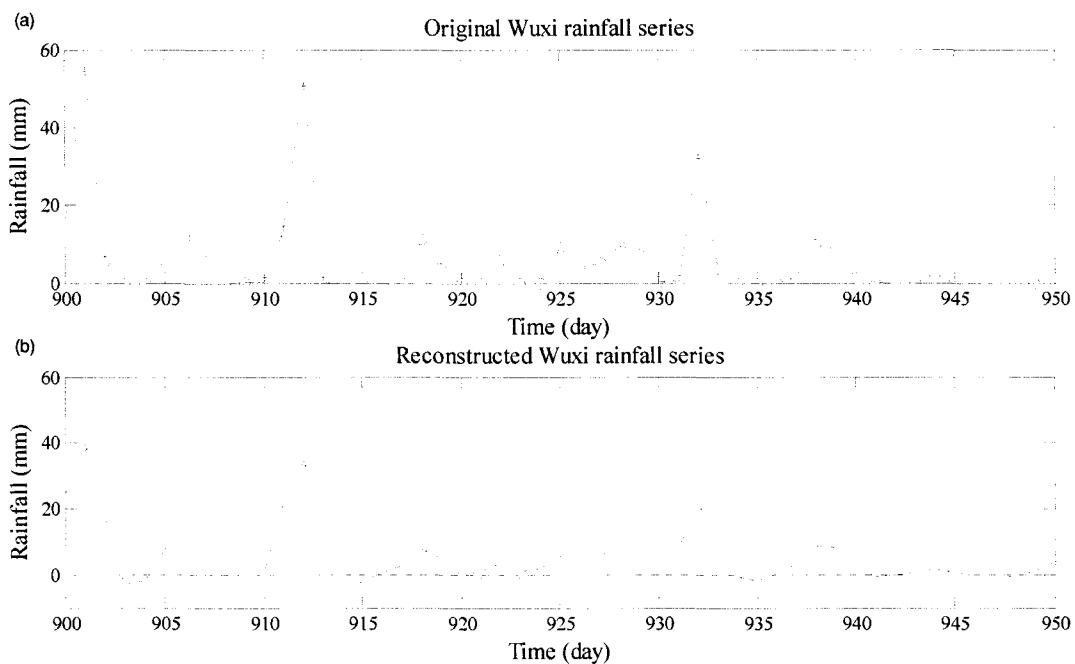


Figure 6.23 Hyetographs (detail; from 900th to 950th) of Wuxi: (1) the raw rainfall series and (2) reconstructed rainfall series for one-step prediction of ANN.

To investigate the influence of SSA on the ANN's performance, correlation analyses between inputs and output of ANN and ANN-SSA2 are compared using the Wuxi

data and are depicted in Figure 6.24. As the input and output series in ANN are both the raw rainfall series, the cross-correlation analysis is equivalent to the autocorrelation analysis of the raw rainfall series. At all three prediction horizons, cross-correlation coefficients (CCs) between reconstructed inputs by SSA2 and the raw rainfall data are significantly improved at most of lags except for the lag of 3 at one-step lead. It should be recall that the model inputs are the previous five rainfall data for Wuxi. The “starting point” in Figure 6.24 represents the first previous rainfall of five inputs, and the remaining inputs consist of four points after the starting point. It can be observed that the CCF value between each new model input and output are far larger than that between the raw model input and output (seen at the same lag). Therefore, the improvement of a model’s performance by the SSA technique may be owing to the enhancement of the mapping relation of model input and output by deleting noises hidden in the raw signal.

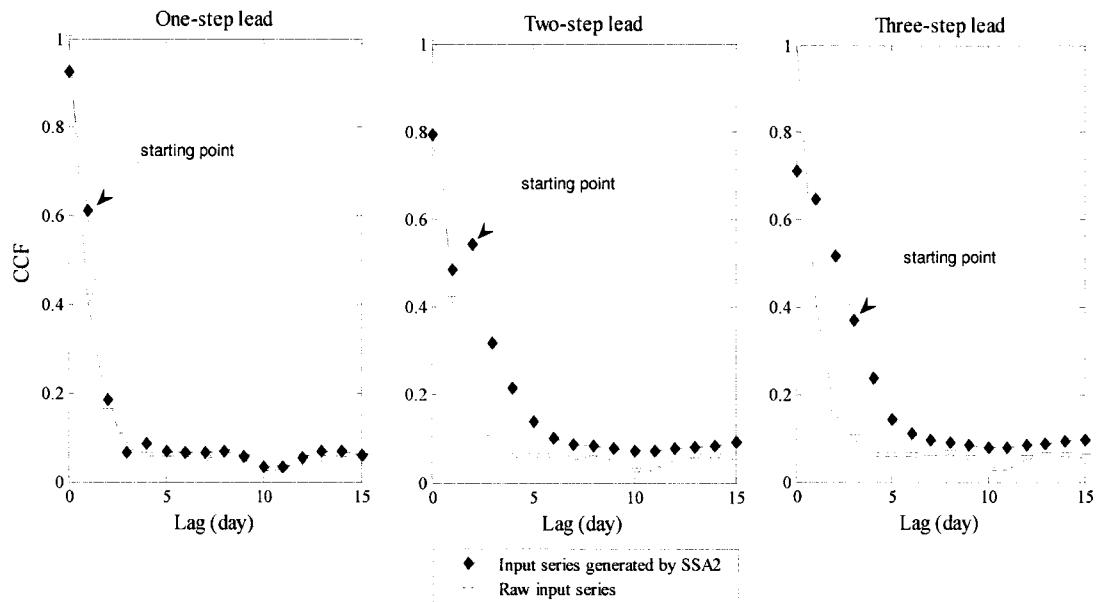


Figure 6.24 CCF between inputs and output for ANN and ANN-SSA2 at three prediction horizons using the Wuxi data

(2) About parameter L in SSA

The parameter L in SSA has a significant impact on the performance of a forecasting model since the optimal p RCs may be different with the change of L .

when using the same forecasting model at the same prediction horizon. The selection of L in this study is based on the interval of [3, 10] in conjunction with an empirical criterion (namely, a particular L is considered as the target only if the singular spectrum under that L can be markedly distinguished). To check the robustness of the empirical method, each L in [3, 10] was examined by the LR model with SSA2 using the Wuxi and Zhenwan data and presented in Table 6.10. As mentioned previously, the target L for Wuxi and Zhenwan are 5 and 7. The RMSE associated with them at each prediction horizon is highlighted in bold (shown in Table 6.10). In terms of Wuxi, the difference between the target RMSE and the minimum RMSE at the same prediction horizon is only 9.2% for one-step prediction, 1.4% for two-step prediction, 0.0% for three-step prediction, respectively. Regarding Zhenwan, the three values were respectively 5.2%, 7.5%, and 0.0%, respectively. These changes are slight and cannot influence the conclusions drawn previously. Therefore, the empirical method for the present rainfall data should be appropriate.

Table 6.10 RMSE of LR coupled with SSA2 using various L

| Watershed | Prediction horizons | L in SSA | | | | | | | |
|----------------|---------------------|------------|-------|-------------|------|-------------|------|------|------|
| | | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Wuxi | | | | | | | | | |
| | 1 | 6.13 | 5.94 | 6.01 | 6.41 | 5.83 | 5.81 | 5.61 | 5.51 |
| | 2 | 7.79 | 7.62 | 7.73 | 8.14 | 7.71 | 7.75 | 7.62 | 7.66 |
| | 3 | 11.76 | 8.61 | 8.40 | 9.04 | 9.23 | 8.72 | 8.56 | 8.62 |
| Zhenwan | | | | | | | | | |
| | 1 | 7.74 | 7.49 | 7.31 | 7.29 | 7.19 | 6.99 | 7.08 | 6.84 |
| | 2 | 10.27 | 8.42 | 9.04 | 8.19 | 7.99 | 7.67 | 7.43 | 7.61 |
| | 3 | 11.28 | 10.66 | 10.06 | 9.33 | 8.81 | 8.91 | 9.42 | 9.28 |

6.5 Summary

This chapter has presented applications of data-driven models coupled with data preprocessing techniques to monthly and daily rainfall forecasts. To achieve more accurate predictions, the optimal forecasting model amongst four candidate data-driven models (LR, K-NN, ANN and MANN) is sought from the perspectives

of model inputs and data preprocessing techniques. Main conclusions can be highlighted as follows:

As far as determination of model inputs is concerned, LCA is identified as the preferred approach among all seven input techniques because of its relative ease in operation and computations, and comparable forecast capability in the context of ANN.

In the normal mode, MANN stands out from the other three models for both monthly and daily rainfall series forecasting. It is seen that the two monthly rainfall series are able to be reasonably forecasted by the four models whereas all four models except for MANN have failed to simulate each daily rainfall series with obvious lag effects.

Amongst all three data preprocessing techniques, the effects of MA and PCA (namely, PCA1 and PCA2) on the improvement of models are negligible. However, SSA has made tremendous contributions on improving all models' performance. Similar to the normal mode, MANN still outperforms the other models. Two methods for filtering of RCs, supervised and unsupervised methods, are proposed and compared. The unsupervised method tends to be better if the number of raw RCs is small. Otherwise, the supervised method is a good alternative.

Proposed forecasting model for rainfall forecasting is therefore MANN coupled with SSA. However, there is still considerable room for improving forecasts in peak values of daily rainfall series.

7 Streamflow Forecasting

Two categories of streamflow forecasts are generally found in the context of data-driven models: one is that the forecasting depends on the flow itself (also termed univariate forecasting), and the other is that the forecasting entails appropriate exogenous variables apart from the flow itself (also termed multivariate forecasting, such as rainfall-runoff transformation). This chapter will present the univariate forecasting for both monthly and daily streamflow series using data-driven models coupled with data preprocessing techniques. The goals of this chapter are:

- (1) To investigate model performance of each of ANN, MANN, ANN-SVR and DSBM;
- (2) To compare SSA and WA as model input filtering techniques.

7.1 Introduction

As an important issue in hydrology time series, much attention has been paid to streamflow series prediction over the past several decades. Numerous data-driven modeling techniques have been proposed for the forecast and simulation of the streamflow series (Carlson et al., 1970; Jain et al. 1999; cannas, et al., 2006; Wang et al, 2006b; Lin et al., 2006). Based on the assumption that hydrological time series is originated from a stochastic process with an infinite number of degrees of freedom, diversified linear models such as Autoregressive and ARMA (Carlson et al., 1970; Salas et al., 1985; Haltiner and Salas, 1988; Yu and Tseng, 1996; Kothyari and Singh, 1999; Huang et al., 2004) have made a great success in the river flow prediction. Meanwhile, chaos theory-based streamflow predictions have been gaining interests of the hydrology community in the past two decades (Jayawardena and Lai, 1994; Jayawardena and Gurung, 2000; Elshorbagy et al., 2002; Sivakumar et al., 2002) where a streamflow series is assumed to be derived from a nonlinear deterministic dynamic system. However, some doubts have been raised in terms of the existence of chaos in hydrologic data (Koutsoyiannis and Pachakis, 1996; Pasternack, 1999;

Schertzer et al., 2002; Wang et al., 2006a).

Describing a streamflow process as either a totally linear stochastic process or fully nonlinear deterministic chaos is not a practical approach since it is in essence an integration of stochastic and deterministic components (Elshorbagy et al., 2002). The model based on either of two assumptions may not be fully appropriate. Recently, various soft computing methods such as ANNs and SVR have been employed to capture the nonlinearity in a streamflow series. A comparison between ANN or SVR and linear modeling methods has been widely reported (Raman and Sunilkumar, 1995; Jain et al., 1999; Abrahart and See, 2002; Thirumalaiah and Deo, 2000; Castellano-Méndez et al., 2004; Kiş, 2005; Lin et al., 2006). Most studies have shown that ANNs outperformed traditional statistical methods. On the other hand, there seems to be no consensus on the performance assessment of ANN or SVR and dynamical system-based method (hereafter referred to as DSBM which is often termed as NLP (nonlinear prediction) when a local forecasting method is adopted). For example, Sivakumar et al. (2002) found that NLP outperformed ANN for short-term river flow predictions. Laio et al. (2003) used NLP and ANN for flood predictions and found that NLP performed slightly better at short forecast time whereas the situation was reversed for longer time. Yu et al. (2004) proposed a scheme that combined chaos theory and SVR to predict the daily discharge where NLP performed the worse than ARIMA and SVR. Addressing comparison between DSBM and other proposed models becomes one of motives of the present study.

It is generally accepted that underlying mechanisms of streamflow generation are likely to be quite different during low, medium and high flow periods. For example, the high flows depend primarily upon the outburst of a heavy storm in the immediate past and geomorphologic properties. Conversely, the low flows are affected primarily by releases from groundwater storage. A single global ANN model cannot predict the high- and low-runoff events satisfactorily (Minns and Hall, 1996). Modular models are therefore proposed where sub-processes are first of all identified and then separate models (also termed local or expert model) are established for each of them (Solomatine and Ostfeld, 2008). Depending on the soft or hard split of training data , different modular models exist. Soft split means that the subset can be overlapped

and the overall forecasting output is the weighted-average of each local model (Zhang and Govindaraju, 2000; Cheng et al., 2006; Shrestha and Solomatine, 2006). On the contrary, there is no overlapping of subset in the crisp split and the final forecast output is explicitly from only one of local models(Corzo and Solomatine, 2007; Jain and Srinivasulu, 2006; See and Openshaw, 2000; Sivapragasam and Liong, 2005; Solomatine and Xue, 2004; Wang et al., 2006b). ANN (or similar methods) has a week ability of extrapolation beyond the range of the data used for training. Therefore, soft split-based modular model may suffer from poor forecasts/predictions without cautious data preprocessing. As far as SVR is concerned, this problem was noticed by Wu et al. (2008).

The hybrid of ANN and SVR is less studied in hydrology. ANN and SVR are mathematically similar when they are used as approximators to map input/output pairs (Vapnik, 1995). Compared with SVR, ANN is very efficient in processing large-size low dimension training samples when a local optimization technique, such as L-M algorithm, is adopted for optimization of weights. In contrast, SVR has a good ability of generalization due to the adoption of structural risk minimization (SRM) for objective function. Therefore, a hybrid method from ANN and SVR (hereafter referred to as ANN-SVR) may be expected to provide a pronounced improvement in the streamflow forecast where the hard split is adopted.

A natural flow series can be viewed as a quasi-periodic signal contaminated by various noises at different flow levels. Therefore, signal decomposition techniques for the purpose of data-preprocessing may be favorable. Two such techniques are known as SSA and WA. Briefly, SSA decomposes a time series into a number of components with simpler structures, such as a slowly varying trend, oscillations and noise. SSA uses the basis functions characterized by data-adaptive nature, which makes the approach suitable for the analysis of some nonlinear dynamics (Elsner and Tsonis, 1996). WA decomposes a time series into a series of linearly independent detail signals and one approximation signal by using discrete wavelet transform with a specific wavelet function such as the Haar wavelet. Mallat (1989) presented a complete theory for wavelet multi-resolution signal decomposition (also mentioned as pyramid decomposition algorithm). Both SSA and WA have been successfully

introduced to the field of hydrology (Lisi et al., 1995; Sivapragasam et al., 2001; Sivapragasam et al., 2007; Marques et al., 2006; Partal and Kişi, 2007). Sivapragasam et al. (2001) established a hybrid model of support vector machine (SVM) with the SSA for predictions of rainfall and runoff. A considerable improvement in the model performance was obtained compared with the raw SVM model. The applications of WA to precipitation and discharge were presented in the work of Partal and Kişi (2007) and Partal and Cigizoglu (2008) respectively, where WA was applied to each input variable. Results indicate that WA is highly promising for improvement of the model performance.

Two main objectives are involved in this chapter: one is to develop ANN-SVR for streamflow forecasting in comparison with ANN, MANN and DSBM; the other is to examine the effect of each model in conjunction with SSA or WA, on improvement of the accuracy of forecasting. SSA and WA as data preprocessing techniques are implemented in two schemes. One is that the raw flow data is first decomposed by SSA or WA, then a new flow series is obtained by filtering, and finally the new flow series is used to generate model inputs. This scheme is denoted by SSA-A or WA-A. The other scheme is denoted by SSA-B or WA-B where model inputs are first determined, then each input series is decomposed by SSA or WA, and finally new input series is generated by removing noise components for each input series. To ensure wider applications of the conclusions, four streamflow series from China, including two monthly and two daily, are explored. This chapter is organized in the following manner. Followed by the section of introduction, four streamflow series are described in Section 7.2. Section 7.3 presents modeling methods and their applications to four streamflow series. The optimal model is identified and the implementation of SSA and WA are described. In Section 7.4, main results are shown along with the necessary analysis. Section 7.5 gives the summary of this chapter.

7.2 Study Area and Data

All four streamflow data are gathered from the watershed of Yangtze River, including two monthly flow series and two daily flow series.

One of the monthly flow series is termed “Danjiangkou” since the hydrology station providing data is located at the Danjiangkou Dam site which is in Hubei Province. The Danjiangkou Dam is situated at the upstream of Han River which is the largest first-order tributary of the Yangtze River and has an average yearly discharge of $1203 \text{ m}^3/\text{s}$. The catchment area controlled by the hydrology station is around $9.5 \times 10^4 \text{ km}^2$. The data period is from January 1930 to December 1981. The other monthly series is “Xiangjiaba” which has an average yearly discharge of $4538 \text{ m}^3/\text{s}$ according to the statistical period from January 1940 to December 1997. The hydrology station is at the main channel of the Yangtze River and is near the Xiangjiaba Dam site which is located in Sichuan Province. The basin area contributes to the streamflow series is around $4.6 \times 10^5 \text{ km}^2$.

Two daily flow series are from river basins of Daning and Lushui. The Daning River, a first-order tributary of the Yangtze River, is located in the northeast of Chongqing city (Figure 6.1). The flow data were collected at “Wuxi” hydrology station which is at the upper and middle streams of the Daning watershed. The drainage area controlled by the Wuxi station is around $2\ 000 \text{ km}^2$. The collected flow data spans from January 1, 1988 to December 31, 2007 (hereafter the flow is referred to as Wuxi). The Lushui River is also a first-order tributary of the Yangtze River (Figure 7.1). The flow data from Lushui River were collected at “Chongyang” hydrology station (hereafter the flow data is referred to as Chongyang). The drainage area controlled by the station is around $1\ 700 \text{ km}^2$. The data period covers a 4-year long duration (January 1, 2004- December 31, 2007).

Figure 7.2 shows the hydrographs of four streamflow series. A linear fit to each hydrograph is denoted by the dashed line. These linear fits (except for Chongyang) are close to horizontal, which indicates that each series is characterized by a good consistency and appear stationary in a weak sense. There is a decreasing trend in the Chongyang data due to the occurrence of a dry period and a shortage data period.

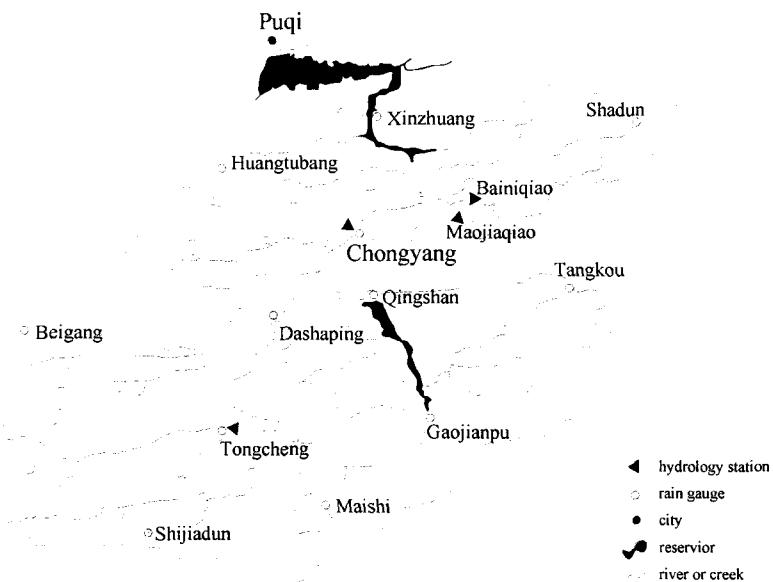


Figure 7.1 Map of the Lushui watershed

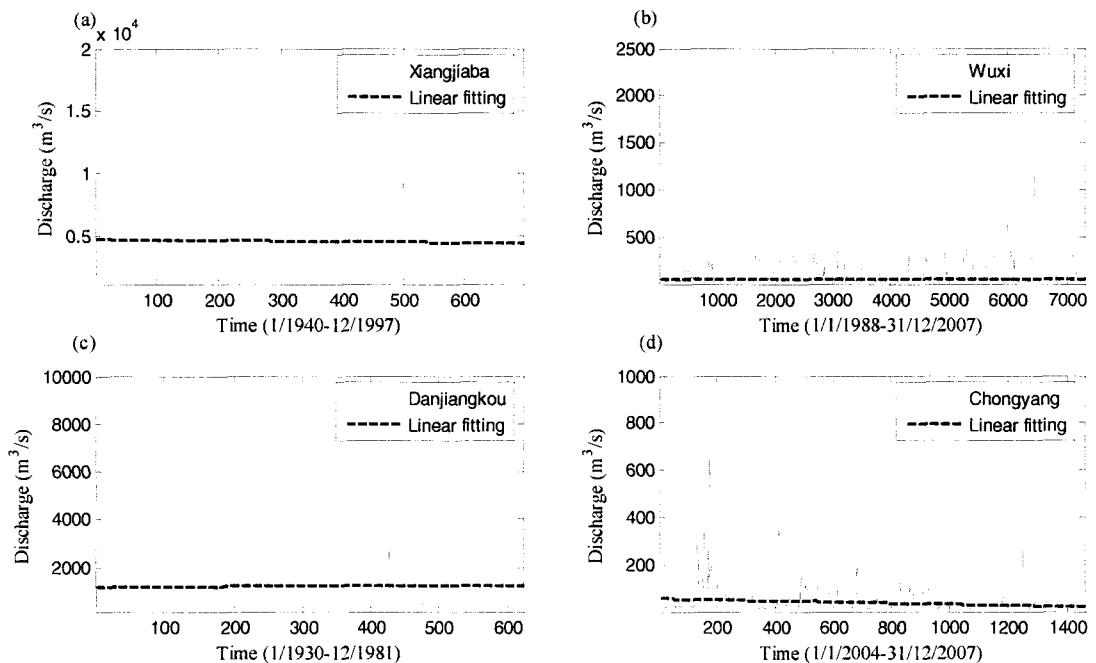


Figure 7.2 Streamflow series of (a) Xiangjiaba, (b) Wuxi, (c) Danjiangkou, and (d) Chongyang

The dataset of each series is partitioned into three data subsets as training set, cross-validation set and testing set: the first half of the entire data as training set and the first half of the remaining data as cross-validation set and the other half as testing set. The training set serves the model training and the testing set is used to evaluate the performances of models. The cross-validation set has dual functions: one is to

implement an early stopping approach so as to avoid overfitting of the training data, and another is to select some best predictions from a large number of ANN's runs. In the present study, 10 best predictions are selected from total 20 ANN's runs.

Table 7.1 presents pertinent information about watersheds and some descriptive statistics of the original data and three subsets, including mean (μ), standard deviation (S_x), coefficient of variation (C_v), skewness coefficient (C_s), minimum (X_{\min}), and maximum (X_{\max}). As shown in Table 7.1, the training set cannot fully include the cross-validation and testing data. It is recommended that all data should be scaled to the interval [-0.9, 0.9] or similar intervals instead of [-1, 1] when ANN employs the hyperbolic tangent sigmoid functions as transfer functions in the hidden layer and output layer.

Table 7.1 Pertinent information for four watersheds and the streamflow data

| Watershed and datasets | Statistical parameters | | | | | | Watershed area and data period |
|------------------------|------------------------|----------------------|-------|-------|---------------------------|---------------------------|--------------------------------|
| | μ (m^3/s) | S_x (m^3/s) | C_v | C_s | X_{\min} (m^3/s) | X_{\max} (m^3/s) | |
| Xiangjiaba | | | | | | | |
| Original data | 4538.2 | 3671.5 | 1.24 | 1.11 | 1100 | 18700 | Area: |
| Training | 4736.4 | 3892.3 | 1.22 | 1.11 | 1130 | 18700 | $4.6 \times 10^5 \text{ km}^2$ |
| Cross-validation | 4146.1 | 3259.6 | 1.27 | 1.09 | 1110 | 15300 | Data period: |
| Testing | 4533.9 | 3592.7 | 1.26 | 1.00 | 1100 | 15000 | Jan., 1940- Dec., 1997 |
| Danjiangkou | | | | | | | |
| Original data | 1203.3 | 1285.1 | 0.94 | 2.47 | 139 | 9010 | Area: |
| Training | 1205.2 | 1352.6 | 0.89 | 2.49 | 140 | 8800 | $9.5 \times 10^4 \text{ km}^2$ |
| Cross-validation | 1292.7 | 1405.3 | 0.92 | 2.50 | 139 | 9010 | Data period: |
| Testing | 1110.0 | 987.9 | 1.12 | 1.58 | 159 | 4980 | Jan., 1930- Dec., 1981 |
| Wuxi | | | | | | | |
| Original data | 61.9 | 112.6 | 0.6 | 7.2 | 6.0 | 2230 | Area: |
| Training | 60.6 | 95.6 | 0.6 | 5.9 | 7.6 | 1530 | $2\ 000 \text{ km}^2$ |
| Cross-validation | 60.7 | 132.2 | 0.5 | 8.3 | 6.0 | 2230 | Data period: |
| Testing | 66.0 | 122.1 | 0.5 | 6.3 | 10.1 | 1730 | Jan., 1988- Dec., 2007 |
| Chongyang | | | | | | | |
| Original data | 39.1 | 54.8 | 0.7 | 6.4 | 2.1 | 881 | Area: |
| Training | 48.1 | 70.1 | 0.7 | 5.5 | 6.9 | 881 | $1\ 700 \text{ km}^2$ |
| Cross-validation | 35.6 | 33.7 | 1.1 | 2.3 | 4.4 | 226 | Data period: |
| Testing | 24.5 | 25.7 | 1.0 | 5.1 | 2.1 | 310 | Jan., 2004- Dec., 2007 |

7.3 Applications of Models

7.3.1 Implementation framework of streamflow forecasting

Figure 7.3 illustrates the framework of streamflow forecasting where four prediction models can be implemented in two modes: without/with SAA and WA (dashed box). Seven acronyms in the box “methods for model inputs” represent different methods and have been explained in Chapter 6.

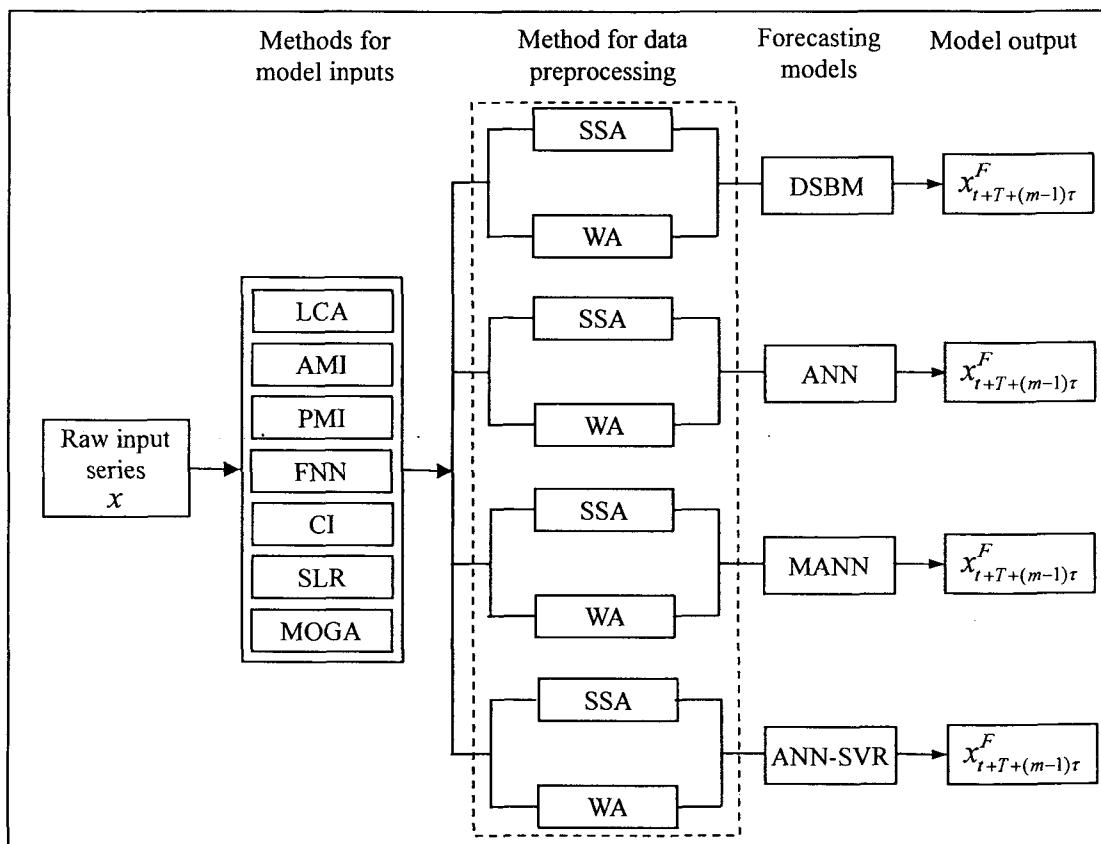


Figure 7.3 Implementation framework of streamflow forecasting models
with/without data preprocessing

7.3.2 Evaluation of model performances

Three measures used in chapter 6, CE, RMSE, and PI, are adopted for the present study as follows:

$$CE = 1 - \sum_{i=1}^n (y_i - \hat{y}_i)^2 / \sum_{i=1}^n (y_i - \bar{y})^2 \quad (7.1)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (7.2)$$

$$PI = 1 - \sum_{i=1}^n (y_i - \hat{y}_i)^2 / \sum_{i=1}^n (y_i - y_{i-l})^2 \quad (7.3)$$

In these equations, n is the number of observations, \hat{y}_i stands for forecasted flow, y_i represents observed flow, \bar{y} denotes average observed flow, and y_{i-l} is the flow estimate from a so-called naïve model (or termed persistence model) that basically takes the last flow observation (at time i minus the lead time l) as a prediction. The value 1 of CE or PI stands for a perfect fit. A small value of PI may imply the occurrence of the lagged prediction.

7.3.3 Determination of model inputs

ANN, equipped with the L-M training algorithm and hyperbolic tangent sigmoid transfer functions, is used as a baseline model to examine seven input methods in terms of RMSE. It is worth noting that ANN, MANN and ANN-SVR are fed by same model inputs.

Table 7.2 presents the results of ANN with different model inputs except for CI because four flow series cannot be identified as chaotic (as depicted in Figure 7.4 where the correlation dimension does not display the property of convergence). These results were based on one-step lead prediction and letting X_{t+1} be the target value at one-step prediction horizon. It can be seen from Table 7.2 that there is only a slight difference between the RMSE from LCA and the minimal RMSE in terms of each experiment. The LCA method is therefore employed in this study since it can significantly reduce the effort and computational time in developing an ANN model.

Table 7.2 Comparison of methods to determine mode inputs using ANN model

| Watershed | Methods | τ | m | Effective inputs ^a | Identified ANN | RMSE |
|--------------------|---------|--------|-----|----------------------------------|----------------|--------|
| Xiangjiaba | | | | | | |
| | LCA | 1 | 20 | the last 12 | (12-9-1) | 1436.3 |
| | AMI | 1 | 12 | except for Xt-8 | (11-10-1) | 1473.9 |
| | PMI | 1 | 12 | Xt-11,t-10 | (2-6-1) | 1725.1 |
| | FNN | 1 | 20 | the last 7 | (7-9-1) | 1441.5 |
| | CI | 3 | 20 | nil | | |
| | SLR | 1 | 12 | Xt,t-1,t-2,t-3,t-7,t-9,t-10,t-11 | (8-7-1) | 1484.6 |
| | MOGA | 1 | 12 | Xt,t-1,t-2,t-3,t-4,t-11 | (6-5-1) | 1410.9 |
| Danjiangkou | | | | | | |
| | LCA | 1 | 20 | the last 12 | (12-8-1) | 755.8 |
| | AMI | 1 | 12 | except for Xt-2,t-8 | (10-10-1) | 805.5 |
| | PMI | 1 | 12 | Xt-11,t,t-9,t-4 | (4-8-1) | 766.3 |
| | FNN | 1 | 20 | last 6 | (6-3-1) | 827.2 |
| | CI | 3 | 20 | nil | | |
| | SLR | 1 | 12 | Xt,t-4,t-7,t-9,t-11 | (5-4-1) | 757.2 |
| | MOGA | 1 | 12 | Xt-11,t-10,t-6,t | (4-1-1) | 735.3 |
| Wuxi | | | | | | |
| | LCA | 1 | 20 | the last 5 | (5-3-1) | 82.2 |
| | AMI | 1 | 13 | the last 13 | (13-7-1) | 83.2 |
| | PMI | 1 | 13 | except for Xt-12 | (12-10-1) | 83.8 |
| | FNN | 1 | 20 | the last 13 | (13-9-1) | 82.6 |
| | CI | 4 | 20 | nil | | |
| | SLR | 1 | 13 | Xt,t-1,t-2,t-4,t-7,t-12 | (6-4-1) | 81.5 |
| | MOGA | 1 | 13 | Xt-10,t-7,t-1,t | (4-10-1) | 82.2 |
| Chongyang | | | | | | |
| | LCA | 1 | 20 | the last 4 | (4-8-1) | 19.7 |
| | AMI | 1 | 12 | except for Xt-10,t-9,t-6,t-5 | (8-5-1) | 20.4 |
| | PMI | 1 | 12 | Xt | (1-8-1) | 20.0 |
| | FNN | 1 | 20 | the last 9 | (9-8-1) | 21.1 |
| | CI | 3 | 20 | nil | | |
| | SLR | 1 | 12 | Xt-5,t-3,t-2,t-1,t | (6-10-1) | 20.7 |
| | MOGA | 1 | 12 | Xt-11,t | (2-6-1) | 21.2 |

Note:^a for the convenience of marking effective inputs, “Xt, t-1”stands for Xt, Xt-1; ^b effective inputs from PMI are shown in descending order of priority. The original values in the column of m are empirically set.

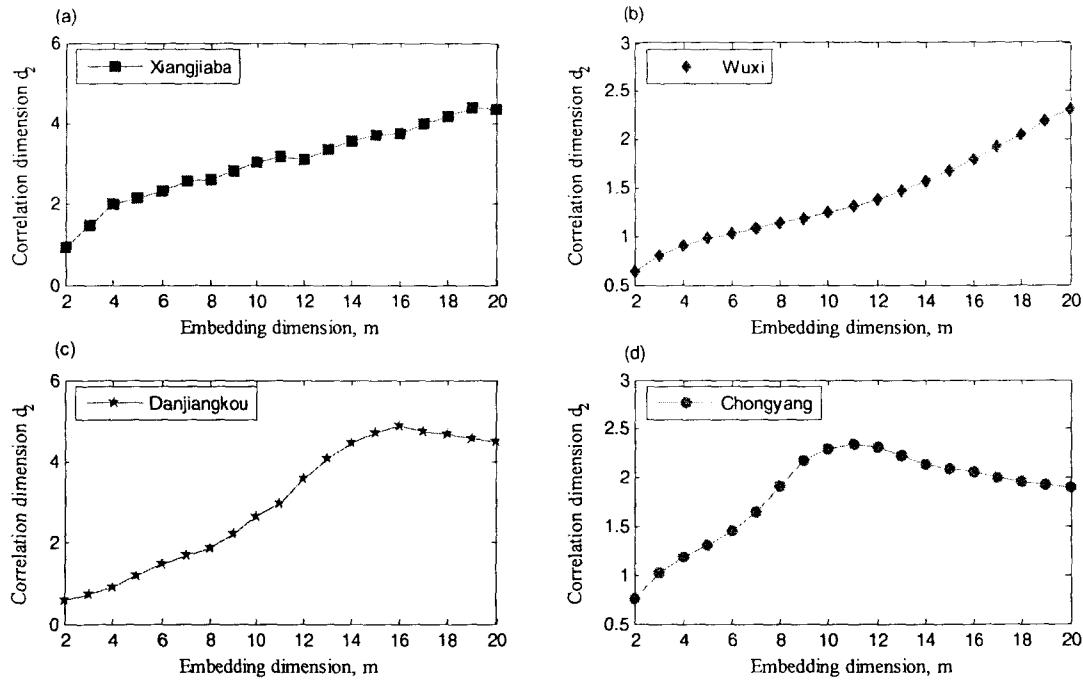


Figure 7.4 Plots of m against d_2 for (a) Xiangjiaba ($\tau = 3$), (b) Wuxi ($\tau = 7$), (c) Danjiangkou ($\tau = 3$), and (d) Chongyang ($\tau = 7$)

Figure 7.5 demonstrates the results of LCA from Wuxi and Chongyang. The model inputs were previous 5-day flows for Wuxi and previous 4 flows for Chongyang because the PACF value decays within the confidence band around at lag 5 for Wuxi and lag 4 for Chongyang, respectively. The model inputs are directly set at the value of 12 for monthly flow series owing to the predominant periodicity being 12 months. Regarding AMI, the effective inputs are also selected based on a 95% confidence limit which is obtained by 200-time bootstraps of training data. The value of τ for the CI method can be defined as: the ACF attains the value of zero or below a small value, or the AMI reaches the first minimum (Tsonis, 1992). The latter is herein adopted as the criterion of the selection of τ . The AMI functions of all four cases are presented in Figure 7.6. Therefore, the values of τ are taken as 3 for Xianjiangba and Danjiangkou, and 7 for Wuxi and Chongyang, respectively.

Regarding the FNN method, a sensitivity analysis of FNNP on R_{tol} (refer to chapter 6 for details) shows that 15 is appropriate value of R_{tol} when τ is set at the value of 1 and R_{tol} varies from 10 to 30 with a step size of 5. In terms of the criterion where

the target m is associated with the first FNNP at less than 1%, the selected m is 7 for Xiangjiaba, 6 for Danjingkou, 13 for Wuxi, and 9 for Chongyang, respectively.

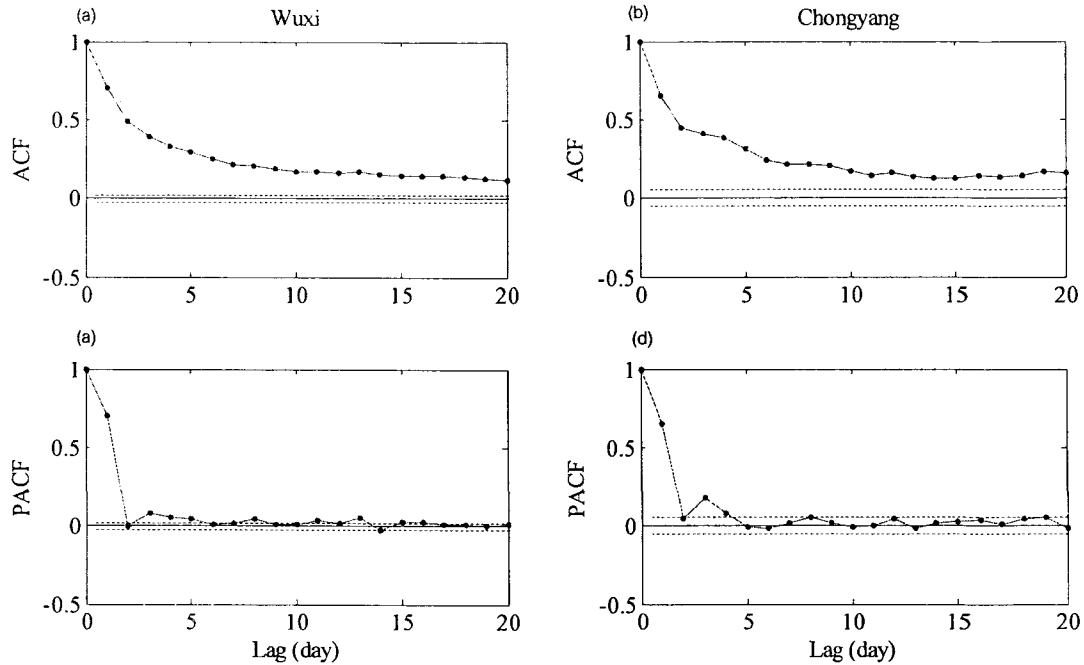


Figure 7.5 Plots of ACF and PACF of the flow series with the 95% confidence bounds (the dashed lines), (a) and (c) for Wuxi, and (b) and (d) for Chongyang

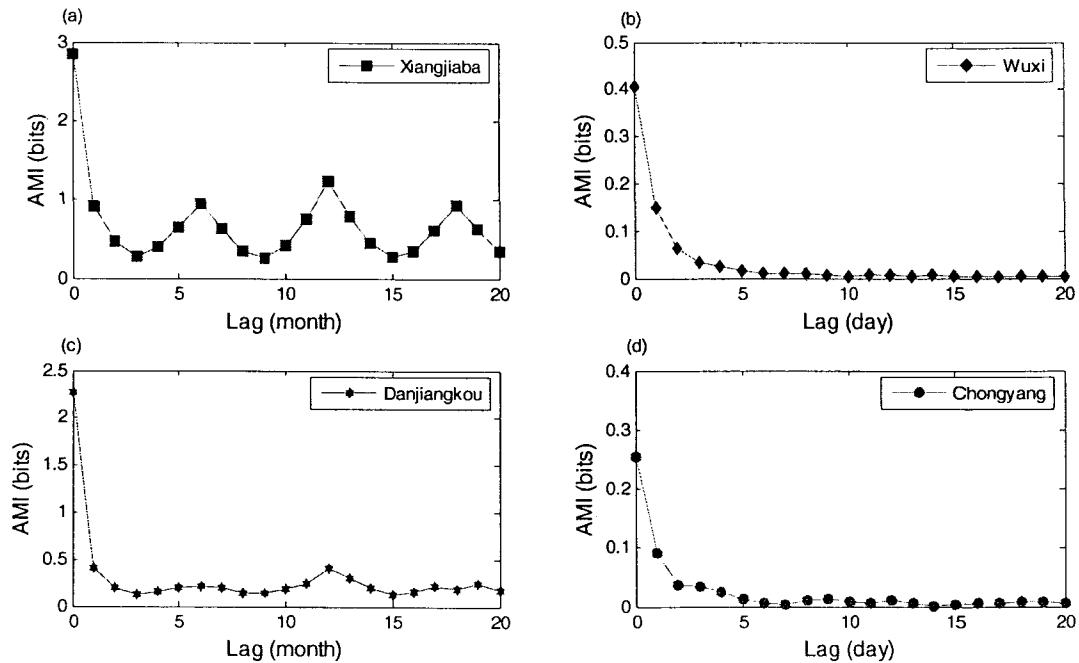


Figure 7.6 Plots of AMI for (a) Xiangjiaba, (b) Wuxi, (c) Danjiangkou, and (d) Chongyang

7.3.4 Identification of models

Four models of DSBM, ANN, MANN, and ANN-SVR are employed for the univariate streamflow series. Descriptions of model applications primarily concentrate on ANN-SVR and DSBM since implementations of ANN and MANN have been described in Chapter 6.

(1) DSBM

The DSBM is based on the assumption that a streamflow time series (say $\{x_1, x_2, \dots, x_N\}$) is derived from a dynamical process. The temporal evolution of the dynamical system can be described in the form of a reconstructed state space where an important parameter pairs (τ, m) is required. Referring to Section 2.2 in Chapter 2, the forecast at time step T can be defined as

$$x_{t+T+(m-1)\tau}^F = f(\mathbf{Y}(t)) + e_t \quad (7.4)$$

where τ is the delay time as a multiple of the sampling period, m is the embedded dimension, and \mathbf{Y}_t represents a state, namely, $\mathbf{Y}_t = \{x_t, x_{t+\tau}, x_{t+2\tau}, \dots, x_{t+(m-1)\tau}\}$, $t = 1, 2, \dots, N - (m-1)\tau$.

Therefore, the first step in the identification of DSBM is to find parameters τ and m . An optimal value of τ should result in the best separation of neighboring states with the minimum embedding state space. The widely used methods for determining the optimal value of τ include the ACF method and the AMI method. The target τ is defined as: the ACF attains the value of zero or below a small value, or the AMI reaches the first minimum (Tsonis, 1992). Figure 7.6 shows that the values of τ are 3 for Xianjiangba and Danjiangkou, and 7 for Wuxi and Chongyang, respectively. In the meantime, two methods, namely CI and FNN, are used to identify the minimal sufficient embedding dimension m . It is well known that the CI method is exclusively used for the check of the existence of chaos in the studied time series. The CI method is therefore not appropriate since the present four flow series have been identified as non-chaotic. According to the determined τ , the relation between FNNP (in the log scale) and $m \in [1, 20]$ for each flow series is depicted in Figure 7.7.

The determined m are 5 for Xiangjiaba and Danjiangkou, 9 for Wuxi, and 8 for Chongyang, respectively if the tolerance for the FNNP close to zero is 1%.

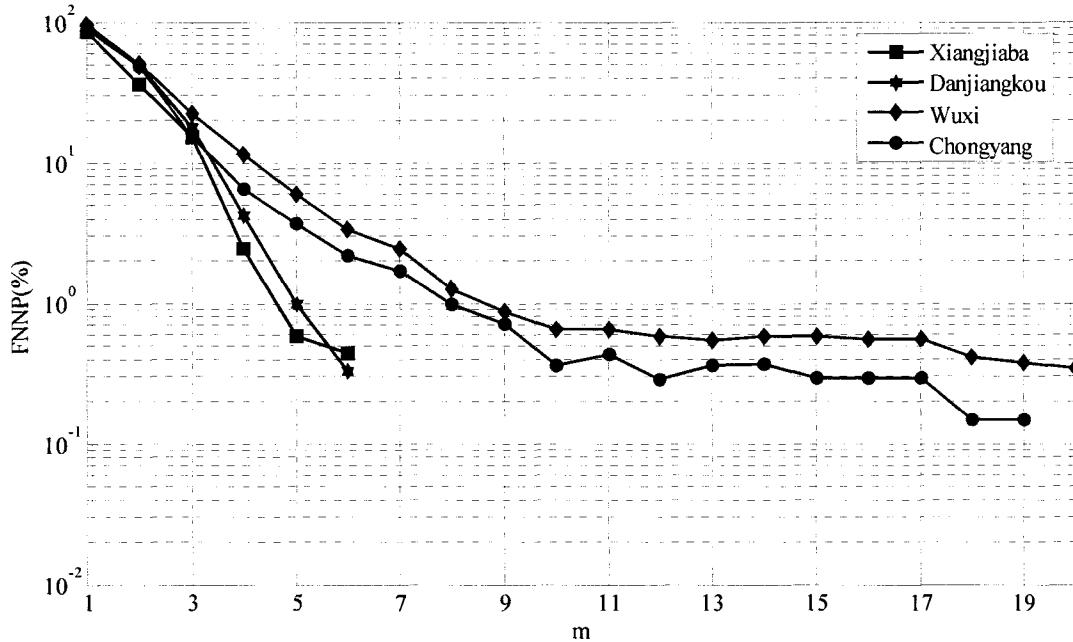


Figure 7.7 Plots of FNNP (in the log scale) with m for all four stream series

Once a state space has been reconstructed with the determined parameter pair (τ, m) , the forecast relies on Eq.(7.4) in order to find the underlying function $f(\bullet)$. As mentioned in the Section 2.2.1 of Chapter 2, both local method and global method can be used to estimate the $f(\bullet)$. The local prediction approach is exactly equivalent to the K-NN method where K 's nearest neighbors of the current state is needed. The value of K is therefore set according to $K = m + 1$, and the forecast is based on the simple weight average of the outputs associated with the K states. Hence, four K 's are set at the values of 6 for both Xiangjiaba and Danjiangkou, 10 for Wuxi, and 9 for Chongyang, respectively. However, ANN is used as a global method to approximate $f(\bullet)$. Based on the L-M training algorithm and the Norm_raw data transformation scheme(see chapter 6), the identified configurations of ANN are 5-7-1 for Xiangjiaba, 5-5-1 for Danjiangkou, 9-6-1 for Wuxi, and 8-1-1 for Chongyang, respectively. The local K-NN method and the global ANN method are respectively denoted by DSBM-L and DSBM-G.

(2) ANN and MANN

As a three-layer perceptron is adopted, the identification of ANN's structure is to optimize the number of hidden nodes h in the hidden layer since the model inputs have been determined by LCA. The optimal size h of the hidden layer was found by systematically increasing the number of hidden neurons from 1 to 10 until the network performance on the cross-validation set could no longer be improved significantly. Based on the L-M training algorithm and the Norm_raw data transformation scheme (seen in Chapter 6), the identified configurations of ANN are 12-9-1 for Xiangjiaba, 12-8-1 for Danjiangkou, 5-3-1 for Wuxi, and 4-8-1 for Chongyang, respectively. The same method is used to identify three local ANNs in MANN. Consequently, the structures of MANN are 12-3/4/1-1 for Xiangjiaba, 12-8/8/8-1 for Danjiangkou, 5-10/10/4-1 for Wuxi, and 4-8/8/5-1 for Chongyang, respectively.

(3) ANN –SVR

Figure 7.8 shows the flow chart of the ANN-SVR model. The hybrid model is based on the crisp data split. A basic idea behind the hybrid model is that the training set is first split into three subsets by FCM, and then each subset is approximated by ANN (or SVR). In terms of a streamflow series, subset1, subset2 and subset3 are expected to denote low, medium and high magnitudes of flows to some extent. In general, the length size of the subset3 should be the smallest. SVR is therefore used in the subset3 since its training speed exponentially increases with the increase of the length size of training data. If the length size of the subset3 is still large, a further partition on the subset3 by FCM has to be processed. Thus, the single SVR is substituted by a modular SVR (MSVR, which consists of k local SVRs) so as to speed up training. It is found that the appropriate k in MSVR is 3 for Xiangjiaba, Danjiangkou and Chongyang, and 6 for Wuxi, respectively. The dashed box denotes that relevant operations on them are optional. For instance, ANN-SVR only conducts predictions in the normal mode if the data preprocessing box is skipped.

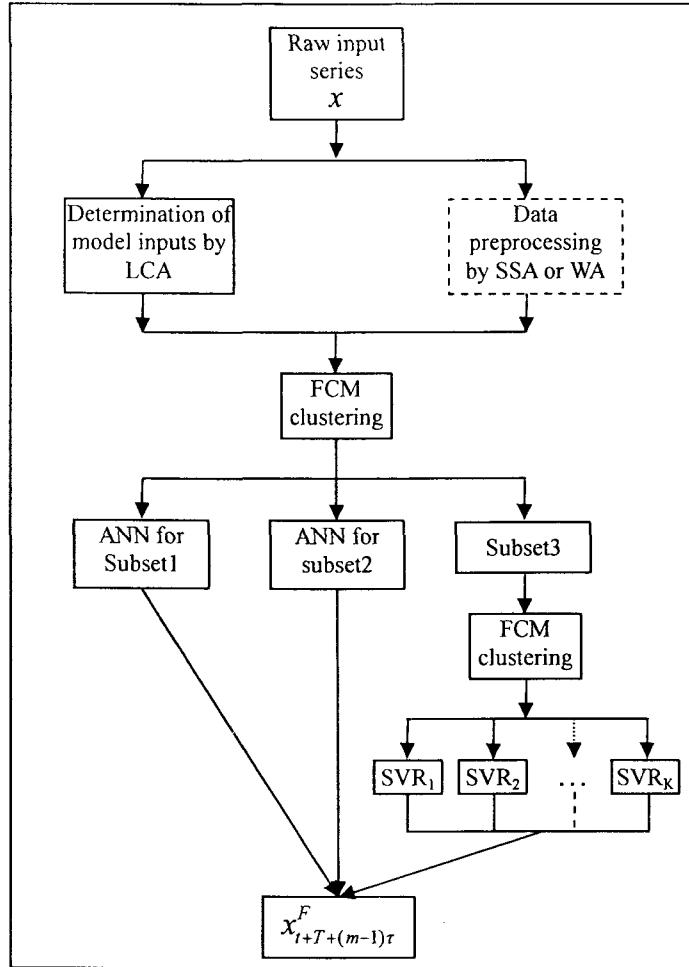


Figure 7.8 Flow chart of the ANN-SVR model

Identification of ANN-SVR consists in determining each local model. Structures of two local ANNs have been obtained in the process of identification of MANN. Regarding MSVR, obtaining optimal α_i and α_i' in Eq. (5.6) for each SVR depends heavily on the triplet parameter (C, ε, σ) when Gaussian RBF is adopted as the kernel function. Due to the lack of any *a priori* knowledge for their bounds, a two-step GA search algorithm is recommended here, which is inspired by a two-step grid search method (*Hsu et al.*, 2003). Firstly, a coarse range search is used to achieve the best region of these three-dimensional grids. In the present study, coarse range partitions for C are $[10^{-2}, 10^0]$, $[10^0, 10^2]$, $[10^2, 5.0 \times 10^2]$, and $[5.0 \times 10^2, 10^3]$. Coarse range partitions for ε are $[10^{-4}, 10^{-3}]$, $[10^{-3}, 10^{-2}]$, $[10^{-2}, 10^{-1}]$, and $[10^{-1}, 10^0]$, and coarse range partitions for σ are $[10^{-3}, 10^{-2}]$, $[10^{-2}, 10^{-1}]$, $[10^{-1}, 10^0]$, and $[10^0, 10^2]$. There are 64 (namely 4^3) grids and one of them is selected as intervals of

parameters for the next step. Then, in the second step a further GA search for the triplet (C, ε, σ) will be carried out in the selected intervals. In order to avoid overfitting of training data, cross-validation data and training data are evaluated concurrently using the fitting degree function of GA (i.e., RMSE), and the weighted average of their fitting degrees is used as the fitting degree of each population in the process of the GA operation. As an example, Table 7.3 shows the optimized triplet (C, ε, σ) in MSVR in the normal mode.

Table 7.3 Calibrated triplet parameters (C, ε, σ) in MSVR with normal mode

| Watershed | Local SVR | Triplet parameters (C, ε, σ) | | |
|--------------------|-----------|---|------------------|-------------------|
| | | 1 | 2 | 3 |
| Xiangjiaba | | | | |
| | SVR1 | 81.9,0.007,1.0 | 65.4,0.035,8.10 | 0.98,0.019,0.21 |
| | SVR2 | 0.97,0.054,1.58 | 9.07,0.069,1.22 | 9.39,0.098,2.69 |
| | SVR3 | 0.91,0.005,1.41 | 9.50,0.009,1.26 | 8.59, 0.0008,2.21 |
| Danjiangkou | | | | |
| | SVR1 | 0.95,0.008,0.14 | 26.0,0.0001,1.0 | 2.32,0.036,0.39 |
| | SVR2 | 0.099,0.019,0.96 | 4.07,0.06,0.006 | 0.82,0.091,0.004 |
| | SVR3 | 0.10,0.0076, 0.62 | 8.99,0.099, 2.03 | 93.7,0.0002,5.66 |
| Wuxi | | | | |
| | SVR1 | 29.8,0.09,1.0 | 0.99,0.0007,0.99 | 33.3,0.053,1.07 |
| | SVR2 | 8.98,0.0084,1.04 | 0.99,0.0008,0.99 | 26.1,0.010,1.29 |
| | SVR3 | 97.7,0.478,1.05 | 20.9, 0.53, 4.28 | 25.24,0.155,0.96 |
| | SVR4 | 38.9,0.155,1.96 | 87.2,0.357,4.15 | 46.2, 1.0, 2.08 |
| | SVR5 | 86.6,0.116,2.26 | 13.1,0.581,1.05 | 19.7,0.213,3.26 |
| | SVR6 | 15.9,0.524,1.92 | 34.6,0.169,1.16 | 67.7,0.54,1.07 |
| Chongyang | | | | |
| | SVR1 | 9.98,0.056,4.0 | 9.23,0.021,1.66 | 0.09,0.0008,5.48 |
| | SVR2 | 2.81,0.0005,1.33 | 9.59,0.031,0.60 | 1.58,0.008,9.31 |
| | SVR3 | 0.09,0.047,0.97 | 0.92,0.059,1.00 | 0.56,0.020,1.01 |

7.3.5 Decomposition of streamflow series

(1) SSA

When SSA is used to decompose a streamflow series, the procedure is the same as that in Chapter 6 for decomposition of the rainfall series. A pivotal task in the procedure is to determine the parameter pair (τ, L) which can be empirically identified by a sensitivity analysis of singular spectrum on them. According to the

criterion of selecting L : L can be viewed as the target only if the singular spectrum under the particular L can be markedly distinguished, L is set at the value of 7 for all four streamflow series by visual inspection. The singular spectrum associated with the selected L is highlighted by the dotted solid line in Figure 7.9.

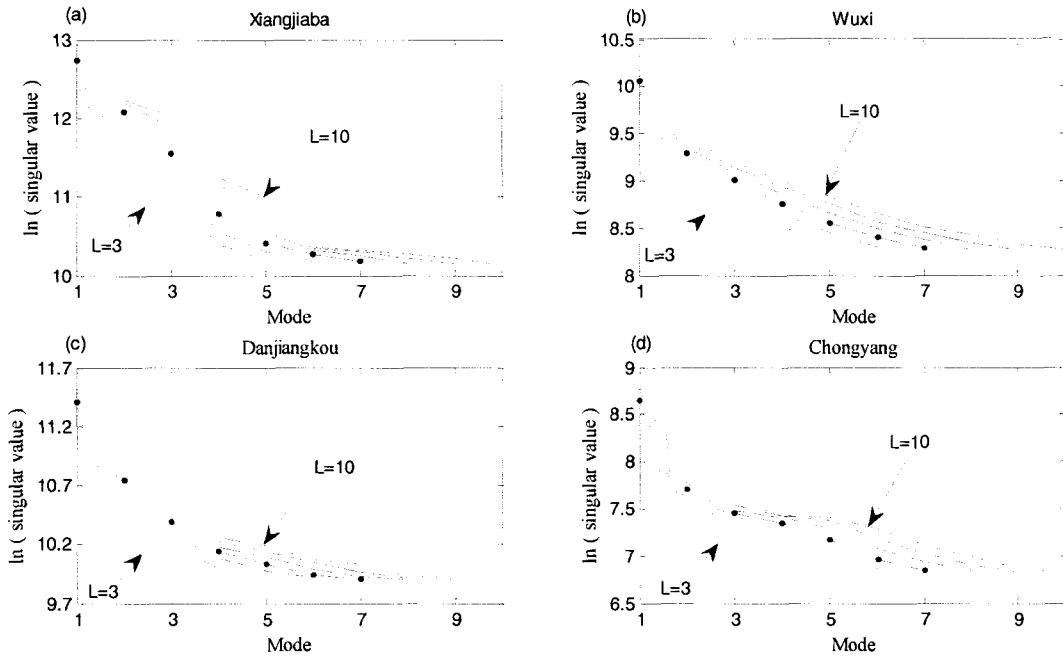


Figure 7.9 Singular Spectrum as a function of lag using varied window lengths L from 3 to 10 for (a) Xiangjiaba, (b) Wuxi, (c) Danjiangkou, and (d) Chongyang

Figure 7.10 presents the results of sensitivity analysis of singular spectrum on the lag time τ using SSA with the determined L . For daily flow series, the singular spectrum can be distinguished only when $\tau=1$. In contrast, the singular spectrum is insensitive to τ for monthly flow series. The final parameter pair (τ, L) in SSA is set $(1, 7)$ for each of four flow series.

Taking Wuxi as an example, Figure 7.11 shows the original flow series and seven RCs excluding the testing data. Each RC is associated with a singular value. For instance, RC1 corresponds to the largest singular value whereas RC7 matches the smallest singular value. The RC1 represents an obvious low-frequency oscillation, which demonstrates a similar mode to the original flow series. In contrast, the RC7 reflects the highest frequency oscillation. The higher is the frequency of one

component, the more possible it is viewed as a noise. Certainty, that whether or not it is a noise is strongly related to the forecasting horizon. For example, one RC identified as a noise at one-step prediction horizon may become a contributing signal at other forecasting horizons.

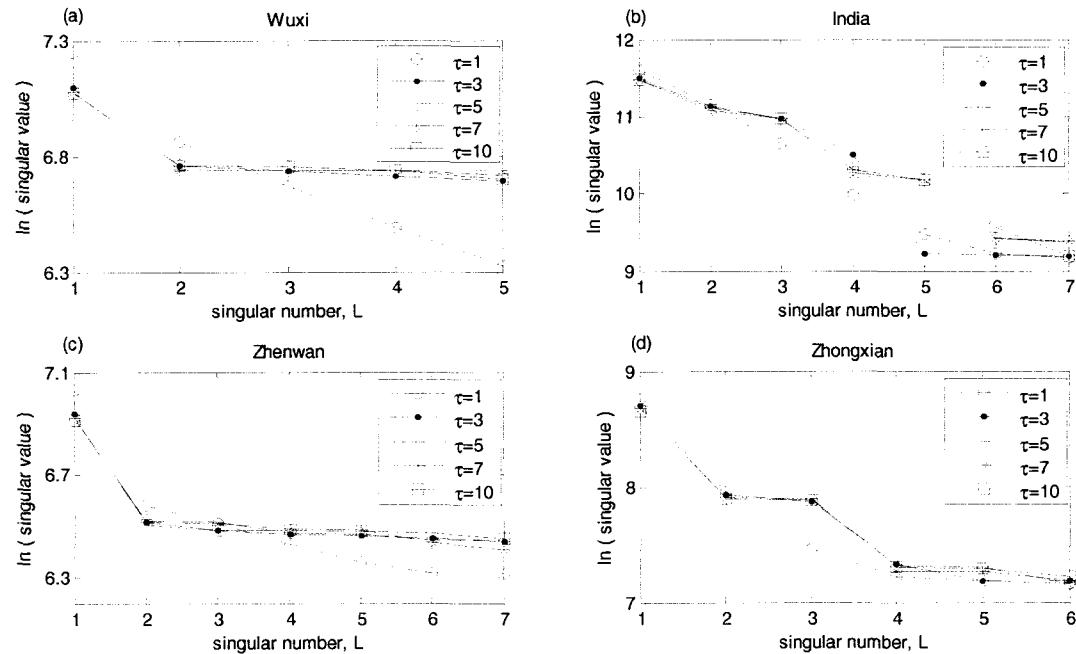


Figure 7.10 Sensitivity analysis of singular Spectrum on varied τ for (a) Xiangjiaba, (b) Wuxi, (c) Danjiangkou, and (d) Chongyang

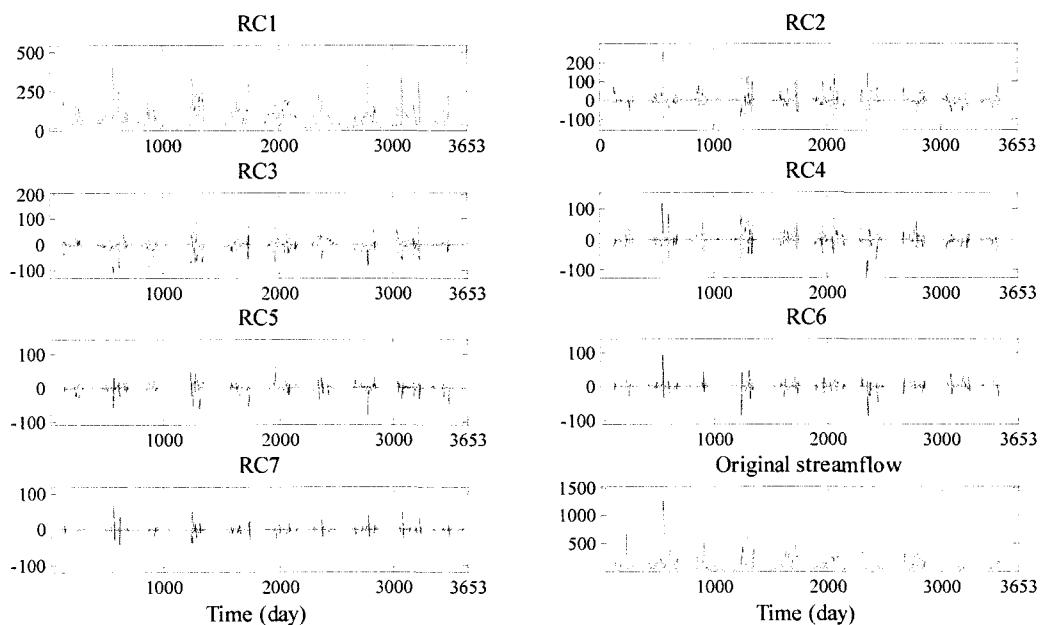


Figure 7.11 RCs and original streamflow series of Wuxi

(2) WA

WA can decompose a streamflow time series into many linearly independent components at different scales (or periods). A low-frequency component generally reflects identification of the signal (such as, trend and periodic signals) whereas a high-frequency component uncovers details. An important issue in the WA is to choose the appropriate number L of scales (referring to Section 4.4 in Chapter 4) so that potential modes hidden in the original series can be extracted. It is recommended that the maximum scale (or period) 2^L be larger than the primary period of the raw series. Continuous wavelet transform (CWT) can reveal the time-frequency characteristic of a time series. Taking the Wuxi series as an example, Figure 7.12 describes the local and global wavelet power spectrum (square of wavelet coefficient) of this streamflow series by CWT. According to the local wavelet power spectrum, it can be observed that the dominant periodic event appears between 256 days and 512 days. The event reflects the annual oscillation of the flow series with an exact period of 365 days from the perspective of common sense. The other noticeable periodic event appears between 512 days and 1024 days, which is associated with a two-year periodic oscillation. In addition, several intermittent oscillations appeared below 128 days along with a marked oscillation between 64 days and 128 days. The global power spectrum in Figure 7.12 (b) is obtained by averaging the local power spectrum with time. Various frequency oscillations are clearly revealed by the power. Therefore, the maximum scale for the Wuxi data is set at 1024 day (2^{10} , namely L is 10) which can include the primary and second oscillations. Likewise, the values of L for Xiangjiaba, Danjiangkou and Chongyang are respectively identified as 7, 6, and 9.

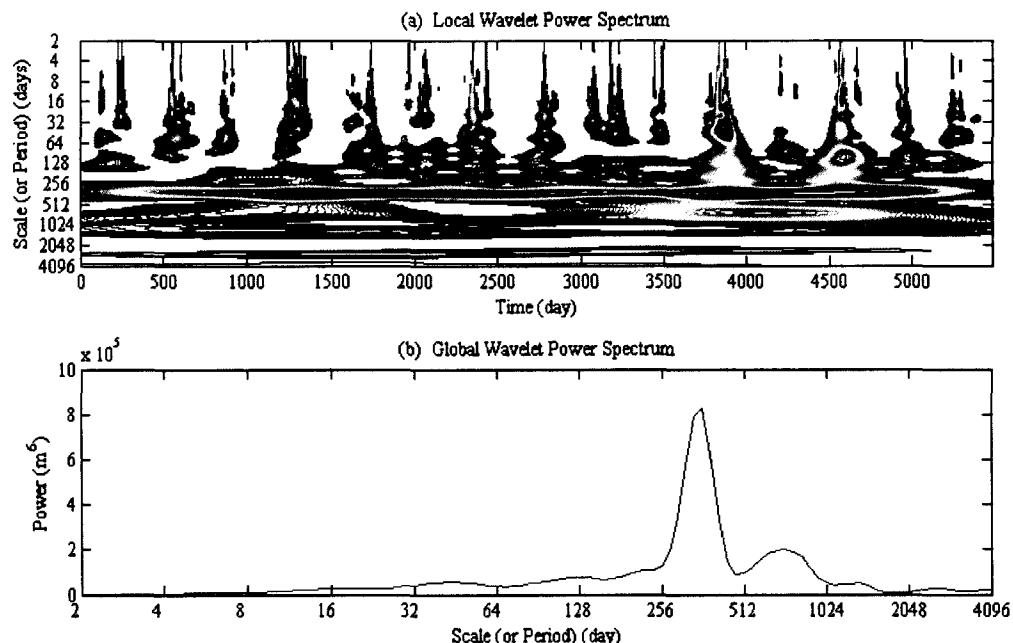


Figure 7.12 (a) Local wavelet power spectrum using the Morlet wavelet function; (b) global wavelet power spectrum from (a) by averaging in time

The Wuxi series is decomposed at 10 wavelet resolution levels ($2^1-2^2-2^3-2^4-2^5-2^6-2^7-2^8-2^9-2^{10}$). Figure 7.13 shows the original flow data (excluding testing data) and 11 wavelet components (10 details and 1 approximation). They were denoted by the notation of DWC together with the power of 2. For instance, DWC1 stands for the component at the scale of 2^1 day and DWC2 represents the component at the scale of 2^2 day, and so on. The DWC11 denotes the approximation component, which was obtained by subtracting all DWCS from the raw flow series. The approximation component reflects the trend of the flow data. As revealed in Figure 7.13, detail components at scales of 2^8 (256 day) and 2^7 (128 day) are characterized by notable periodicity, which partially represent annual oscillation and semi-annual oscillation in the original flow series. Relative weak periodic signals occur at scales of 16 days, 32 days and 64 days. Other high-frequency components tend to capture the details (or noises) of the original flow series. Hence, model inputs can be filtered by deleting some high-frequency DWCS.

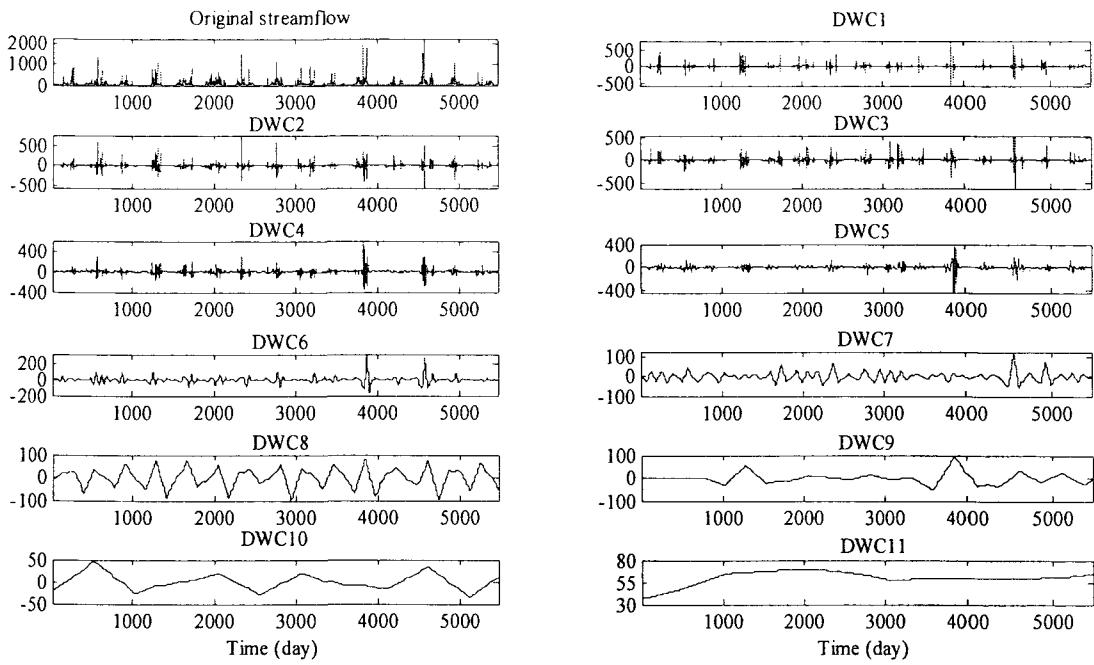


Figure 7.13 DWCs and original streamflow series of Wuxi

Figure 7.14 depicts CCFs between DWCs and the original data of Wuxi. The last plot in this Figure presents the average of CCFs from all DWCs. The average indicates an overall correlation between input and output at various lags (also termed prediction horizons).

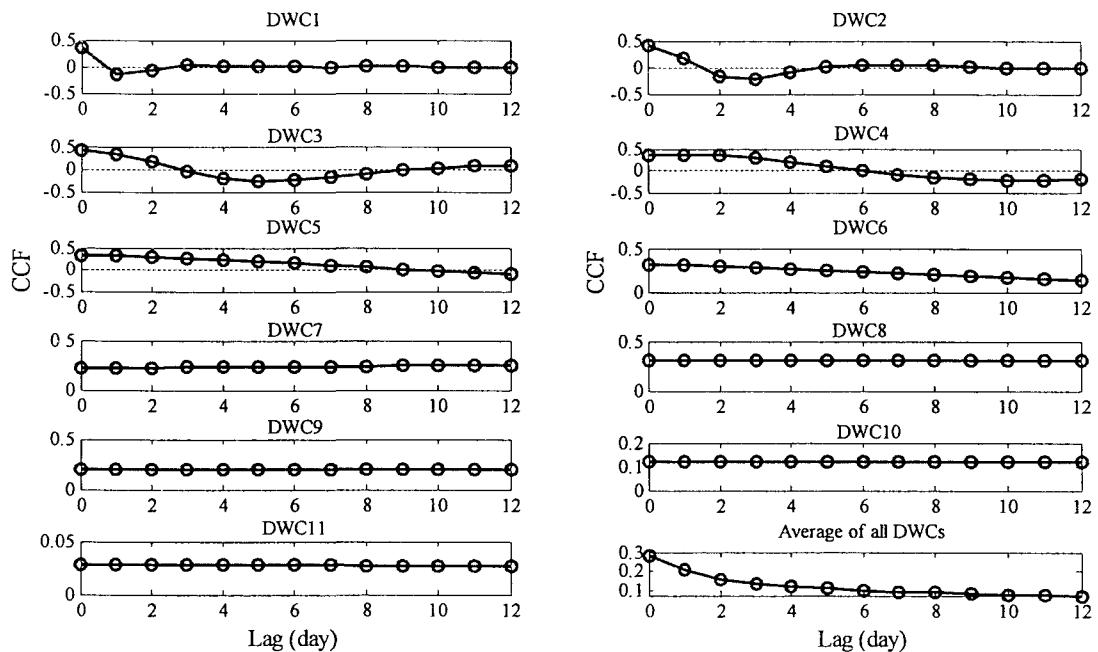


Figure 7.14 Plots of CCF between each DWC and the raw data for Wuxi

7.3.6 Combination of models with SSA or WA

The forecasting models can be coupled with SSA or WA once RCs or DWCs of the flow series have been obtained. As mentioned in the section of introduction, two data preprocessing schemes, scheme A and scheme B, are recommended.

(1) Scheme A

The purpose of scheme A is to compare effects of SSA and WA on the improvement of model performance. In this scheme, a new flow series is firstly generated by filtering of RCs or DWCs, and then the new flow series is used to construct input vectors. As mentioned in Chapter 6, there are unsupervised and supervised filtering techniques. The unsupervised technique is adopted here in view of the number of RCs or DWCs being small. For obtaining more reliable results, three models of ANN, MANN, and ANN-SVR are respectively coupled with SSA or WA. In scheme A, SSA and WA are denoted by SSA-A and WA-A. Hence, when ANN is coupled with SSA in scheme A, it is denoted as ANN-SSA-A.

(2) Scheme B

SSA-B and WA-B represent SSA and WA in -scheme B where the filtering operation is used in each input variable. In general, there are a few model input variables (say m) for any forecast model. If the unsupervised filtering technique is adopted for L RCs or DWCs, there are $2^{m \times L}$ combinations which tend to be huge. Therefore, a supervised filtering technique is adopted for scheme B. ANN is used as the forecasting model. The implementation of the ANN-SSA-A (or ANN-WA-A) model follows the procedure in Figure 6.13. However, taking Wuxi as an example, methodological procedures for ANN-SSA-B (or ANN-WA-B) are different and can be summarized into the following steps.

Step 1: the original time series is decomposed into RCs (or DWCs);

Step 2: the CCFs between each RC (or DWC) and the raw flow data are computed at various lags (such as Figure 7.14 where the lag is up to 12 day);

Step 3: the average CCF curve is obtained by averaging over the CCFs at the same lag. As indicated in Figure 7.14, the average at each lag is positive, which means the overall correlation between each of the last 12 observations and the present observation is positive. In fact, the average represents the correlation between the model input variables and output because the last 5 flow data is used as model inputs for Wuxi;

Step 4: The filtering of RCs (or DWCS) is based on assumption that the combination of components with the same sign in CCF (+ or -) can strengthen the correlation with model output. For each lag (or input variable), the appropriate component are determined by trial and error. In this study, for selecting contributing component to model output, the threshold value varies from 0 to 0.6 with a 0.05 step. For instance, if the average CCF at lag 1 is positive, then these components whose CCF at this lag is less than a set threshold value are deleted. On the contrary, these components whose CCF is larger than the threshold value are deleted if the average CCF at the lag is negative. Depending on trial and error, each model input variable can be filtered. The remaining RCs (or DWCS) for each input variable is associated with the best performance of ANN in terms of RMSE.

7.4 Results and Analysis

Results are presented according to three parts: forecasting from DSBM, ANN, MANN and ANN-SVR in the normal mode; forecasting from ANN, MANN, and ANN-SVR with scheme A, where SSA and WA are compared using the same ANN model; and comparison of schemes, A and B, with ANN as the forecasting model.

7.4.1 Forecasts in normal mode

Table 7.4 demonstrates the results of six models without SSA or WA at three forecasting horizons using four streamflow series data. As the baseline, the naïve

model performs the worst since its prediction is directly equal to the average of one or several nearest observations without considering any mapping relation. The results in each watershed show that DSBM-G is consistently superior to DSBM-L with the increase of the forecasting lead. However, there are similar performances between DSBM-G and ANN although they had different model inputs. Taking the property of instability of the ANN output into account, they can be viewed as alternative to each other. Compared to ANN, two modular models of MANN and ANN-SVR could noticeably improve the accuracy of prediction in all watersheds except for Xiangjiaba. Evidently, the more irregular a streamflow series is, the more obvious the superiority of modular models over ANN tends to be. The close model performance in terms of RMSE, CE and PI between MANN and ANN-SVR has proved the effectiveness of combination of ANN and SVR. Certainly, the performance of ANN-SVR is oftentimes affected by the optimization technique for optimizing the triplet parameters in SVR. Moreover, the training time tends to be longer than that in ANN. Therefore, researchers may prefer MANN although SVR has better capability of generalization in theory.

Table 7.4 Model performances at three forecasting horizons in normal mode

| Watershed | Model | RMSE | | | CE | | | PI | | |
|--------------------|---------|--------|--------|--------|------|-------|-------|------|-------|-------|
| | | 1* | 2* | 3* | 1 | 2 | 3 | 1 | 2 | 3 |
| Xiangjiaba | | | | | | | | | | |
| | Naïve | 2564.6 | 2564.6 | 5256.2 | 0.49 | 0.49 | -1.15 | 0.00 | 0.00 | -0.02 |
| | DSBM-L | 1713.8 | 1713.7 | 1640.3 | 0.77 | 0.77 | 0.79 | 0.55 | 0.82 | 0.90 |
| | DSBM-G | 1490.1 | 1636.7 | 1584.1 | 0.83 | 0.79 | 0.81 | 0.67 | 0.84 | 0.91 |
| | ANN | 1421.1 | 1552.6 | 1539.3 | 0.84 | 0.81 | 0.82 | 0.69 | 0.86 | 0.91 |
| | MANN | 1340.0 | 1544.8 | 1556.9 | 0.86 | 0.82 | 0.81 | 0.73 | 0.86 | 0.91 |
| | ANN-SVR | 1326.5 | 1544.2 | 1525.2 | 0.86 | 0.82 | 0.82 | 0.73 | 0.86 | 0.91 |
| Danjiangkou | | | | | | | | | | |
| | Naïve | 978.5 | 1227.6 | 1441.8 | 0.02 | -0.55 | -1.13 | 0.00 | -0.01 | -0.01 |
| | DSBM-L | 954.8 | 1025.7 | 919.2 | 0.07 | -0.07 | 0.14 | 0.05 | 0.29 | 0.59 |
| | DSBM-G | 777.2 | 877.2 | 739.4 | 0.39 | 0.22 | 0.45 | 0.36 | 0.49 | 0.74 |
| | ANN | 753.1 | 763.2 | 880.7 | 0.42 | 0.41 | 0.21 | 0.41 | 0.61 | 0.62 |
| | MANN | 599.6 | 633.9 | 777.7 | 0.63 | 0.59 | 0.38 | 0.62 | 0.73 | 0.71 |
| | ANN-SVR | 626.8 | 619.8 | 711.6 | 0.60 | 0.61 | 0.48 | 0.59 | 0.74 | 0.75 |
| Wuxi | | | | | | | | | | |

Table 7.4 (continued) Model performances at three forecasting horizons in normal mode

| Watershed | Model | RMSE | | | CE | | | PI | | |
|------------------|---------|-------|-------|-------|------|-------|-------|-------|------|------|
| | | 1* | 2* | 3* | 1 | 2 | 3 | 1 | 2 | 3 |
| Chongyang | Naïve | 90.3 | 127.7 | 139.1 | 0.45 | -0.09 | -0.30 | 0.00 | 0.00 | 0.00 |
| | DSBM-L | 121.0 | 121.9 | 125.3 | 0.02 | 0.01 | -0.05 | -0.78 | 0.10 | 0.19 |
| | DSBM-G | 84.0 | 106.3 | 113.8 | 0.53 | 0.25 | 0.14 | 0.15 | 0.32 | 0.34 |
| | ANN | 81.3 | 104.6 | 111.5 | 0.56 | 0.27 | 0.17 | 0.19 | 0.33 | 0.36 |
| | MANN | 75.7 | 93.7 | 97.1 | 0.62 | 0.41 | 0.37 | 0.30 | 0.46 | 0.51 |
| | ANN-SVR | 80.5 | 94.3 | 102.1 | 0.57 | 0.40 | 0.30 | 0.21 | 0.46 | 0.46 |

* Numbers of “1, 2, and 3” denote one-, two-, and three-step-ahead forecasts

Figure 7.15 shows hydrographs and scatter plots at one-step-ahead prediction from ANN, MANN, and ANN-SVR using the Danjiangkou flow data. These models are not able to appropriately reproduce the real flow process with obvious underestimations on a large number of peak flows although modular models result in a bit improvement of the overall performance.

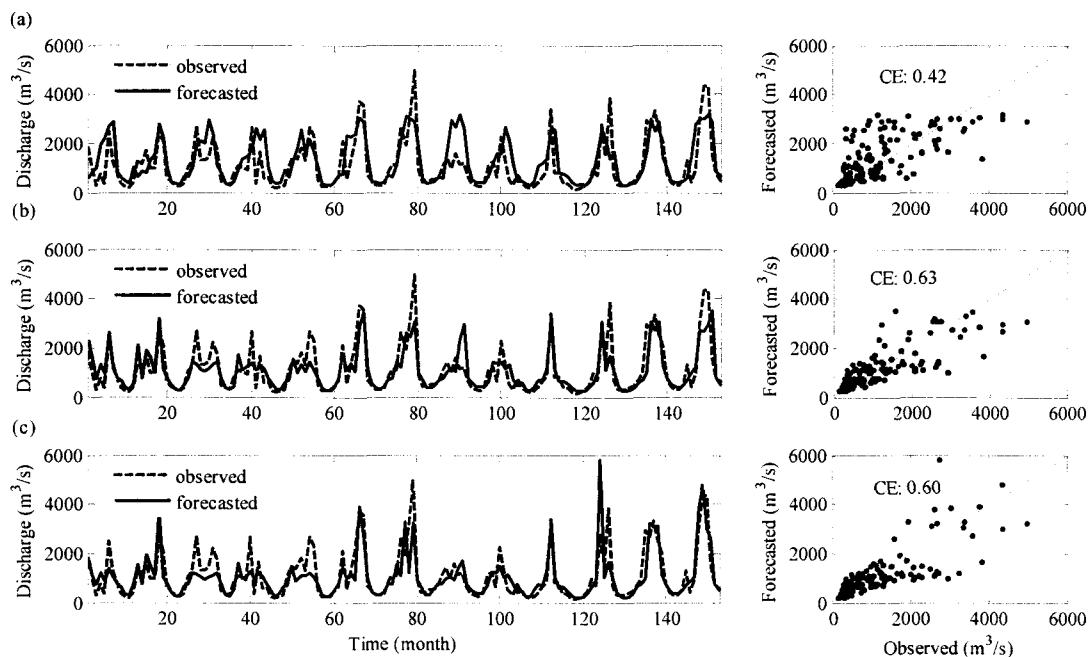


Figure 7.15 Hydrographs and scatter plots of one-step-ahead forecast for Danjiangkou using (a) ANN, (b) MANN, and (c) ANN-SVR

Figure 7.16 shows hydrographs (representative details) and scatter plots at one-step-ahead prediction from ANN, MANN, and ANN-SVR using the Wuxi flow data. Each model has made quite poor forecasts with much underestimation over peak flows. The prediction lag effect can be clearly observed in the hydrograph. To reveal the lag effect, Figure 7.17 presents CCFs at three prediction horizons by the cross-correlation analysis between observed and forecasted flows. The value of CCF at zero lag corresponds to the actual performance (i.e. correlation coefficient) of the model. The target lag is associated with the maximum value of CCF, and is an expression of the mean lag for the entire prediction. Figure 7.17 shows that the lag effect truly occurs in the daily flow series. There are 1, 2, and 3 days lag for Wuxi, which are respectively associated with one-, two-, and three-day-ahead forecasts. In contrast, there are no lagged predictions for monthly streamflow series, such as the Danjingkou data.

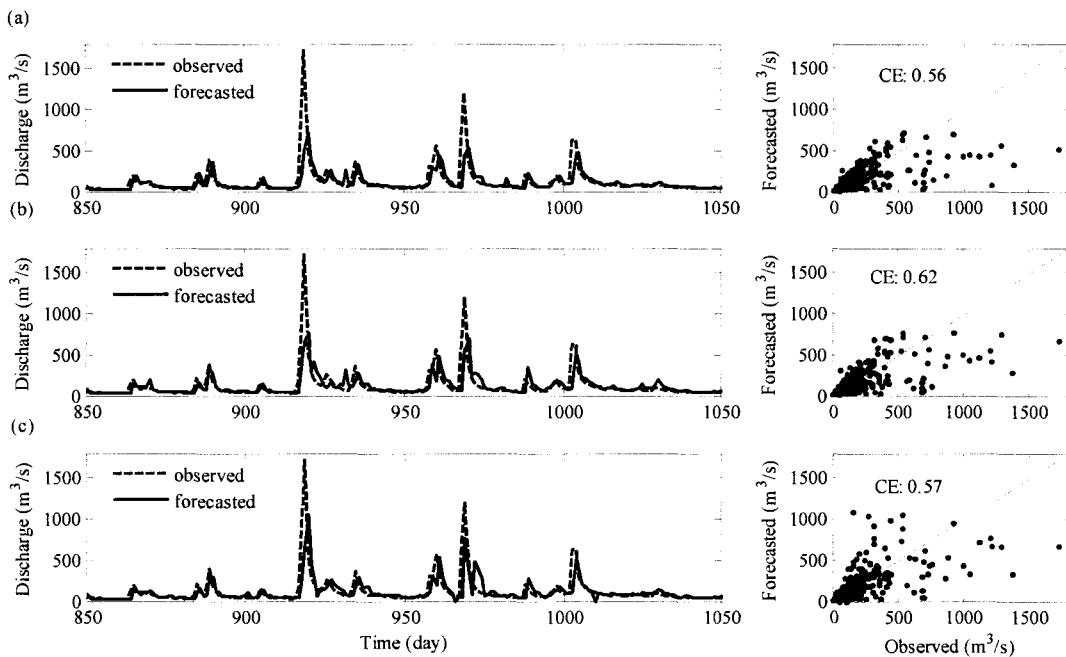


Figure 7.16 Hydrographs and scatter plots of one-step-ahead forecast from (a) ANN, (b) MANN, and (c) ANN-SVR using the Wuxi data

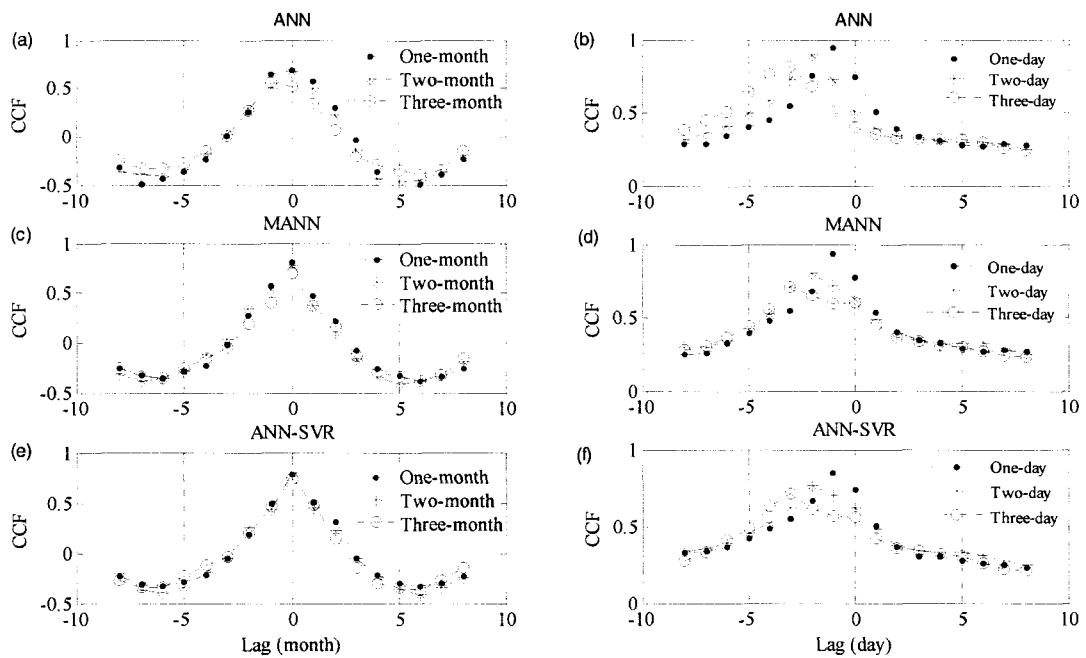


Figure 7.17 CCFs at three forecasting horizons using ANN, MANN, and ANN-SVR
(The left column for Danjiangkou, and the right column for Wuxi)

7.4.2 Forecasts with scheme A

Table 7.5 demonstrates the optimal model input components for three prediction horizons when ANN is coupled with SSA or WA using scheme A as the filtering technique. In terms of RMSE, SSA is consistently superior to WA although the performance of ANN has also been considerably improved by WA. A further comparison is presented in Table 7.6 where two modular models of MANN and ANN-SVR are compared with ANN. Note that, in scheme A, ANN, MANN and ANN-SVR are fed by the same filtered input series. Table 7.6 demonstrates that the three models in the SSA-A mode, are capable of making very accurate forecasts for each prediction horizon, when compared with results in Table 7.4. It can also be observed that MANN-SSA-A and ANN-SVR-SSA-A do not consistently outperform ANN-SSA-A for all three prediction horizons. However, Figure 7.18 shows that MANN-SSA-A can improve the forecasting accuracy in the peak flows of the Wuxi series when both ANN-SSA-A and ANN-WA-A underestimate some peak flows. The hydrograph in Figure 7.18 shows that each model's timing of the peaks is quite good. The scatter plot from ANN-WA-A has quite a large spread with the increase of observed flows, which indicates these flows are not well matched.

Table 7.5 Optimal p RCs (or DWCs) for ANN's input at various prediction horizons

| Watershed | Prediction horizons | ANN-SSA-A | | | | ANN-WA-A | | | |
|--------------------|---------------------|-----------|---------|----------------|---------|----------|------|----------------|--------|
| | | Optimal | p RCs | RMSE | Optimal | p DWCs | RMSE | | |
| Xiangjiaba | | | | | | | | | |
| Xiangjiaba | 1 | 1 | 2 | 4 ^a | 453.7 | 3 | 4 | 5 ^b | 937.0 |
| | 2 | 2 | 5 | 7 | 507.2 | 3 | 4 | 5 | 996.3 |
| | 3 | 2 | 5 | 6 | 496.4 | 3 | 4 | 7 | 1123.1 |
| Danjiangkou | | | | | | | | | |
| Danjiangkou | 1 | 2 | 6 | | 353.6 | 2 | 3 | 4 | 5 |
| | 2 | 2 | 6 | | 312.7 | 3 | 4 | 6 | 592.9 |
| | 3 | 2 | 6 | | 328.5 | 3 | 4 | 5 | 644.7 |
| Wuxi | | | | | | | | | |
| Wuxi | 1 | 1 | 2 | 3 | 4 | 2 | 3 | 4 | 5 |
| | 2 | 1 | 2 | | 50.6 | 3 | 4 | 5 | 11 |
| | 3 | 1 | 2 | | 61.8 | 2 | | | |
| Chongyang | | | | | | | | | |
| Chongyang | 1 | 1 | 2 | 3 | 4 | 1 | 3 | 4 | 5 |
| | 2 | 1 | 2 | 3 | | 3 | 4 | 5 | 10 |
| | 3 | 1 | 5 | 6 | 15.7 | 3 | 4 | 5 | 7 |

Note: ^a numbers of "1, 2, 4" stand for RC, RC2, and RC4, and ^b numbers of "3, 4, 5" stand for DWC3, DWC4, and DWC5.

Table 7.6 Model performances at three forecasting horizons in scheme A

| Watershed | Model | RMSE | | | CE | | | PI | | |
|--------------------|---------------|-------|-------|--------|------|------|------|------|------|------|
| | | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| Xianjiaba | | | | | | | | | | |
| Xianjiaba | ANN-SSA-A | 453.7 | 507.2 | 496.4 | 0.98 | 0.98 | 0.98 | 0.97 | 0.98 | 0.99 |
| | ANN-WA-A | 937.0 | 996.3 | 1123.1 | 0.93 | 0.92 | 0.90 | 0.86 | 0.94 | 0.95 |
| | MANN-SSA-A | 468.8 | 526.1 | 379.1 | 0.98 | 0.98 | 0.99 | 0.97 | 0.98 | 0.99 |
| | ANN-SVR-SSA-A | 655.4 | 556.9 | 469.8 | 0.97 | 0.98 | 0.98 | 0.93 | 0.98 | 0.99 |
| Danjiangkou | | | | | | | | | | |
| Danjiangkou | ANN-SSA-A | 353.6 | 312.7 | 328.5 | 0.88 | 0.90 | 0.89 | 0.87 | 0.94 | 0.95 |
| | ANN-WA-A | 585.5 | 592.9 | 644.7 | 0.66 | 0.66 | 0.59 | 0.65 | 0.77 | 0.81 |
| | MANN-SSA-A | 332.4 | 308.4 | 337.1 | 0.89 | 0.91 | 0.89 | 0.89 | 0.94 | 0.95 |
| | ANN-SVR-SSA-A | 331.6 | 337.6 | 380.5 | 0.89 | 0.89 | 0.86 | 0.89 | 0.93 | 0.93 |
| Wuxi | | | | | | | | | | |
| Wuxi | ANN-SSA-A | 31.9 | 50.6 | 61.8 | 0.93 | 0.83 | 0.74 | 0.87 | 0.84 | 0.80 |
| | ANN-WA-A | 57.9 | 76.4 | 78.8 | 0.77 | 0.61 | 0.58 | 0.59 | 0.64 | 0.68 |
| | MANN-SSA-A | 22.5 | 41.0 | 48.7 | 0.95 | 0.89 | 0.84 | 0.94 | 0.90 | 0.88 |
| | ANN-SVR-SSA-A | 33.2 | 50.2 | 55.7 | 0.93 | 0.83 | 0.79 | 0.87 | 0.85 | 0.84 |
| Chongyang | | | | | | | | | | |
| Chongyang | ANN-SSA-A | 7.9 | 11.2 | 15.7 | 0.96 | 0.81 | 0.63 | 0.86 | 0.83 | 0.72 |
| | ANN-WA-A | 15.4 | 17.3 | 16.1 | 0.64 | 0.55 | 0.61 | 0.68 | 0.59 | 0.70 |
| | MANN-SSA-A | 7.3 | 10.2 | 15.7 | 0.92 | 0.84 | 0.63 | 0.88 | 0.86 | 0.72 |
| | ANN-SVR-SSA-A | 7.8 | 9.5 | 14.9 | 0.91 | 0.86 | 0.67 | 0.86 | 0.88 | 0.75 |

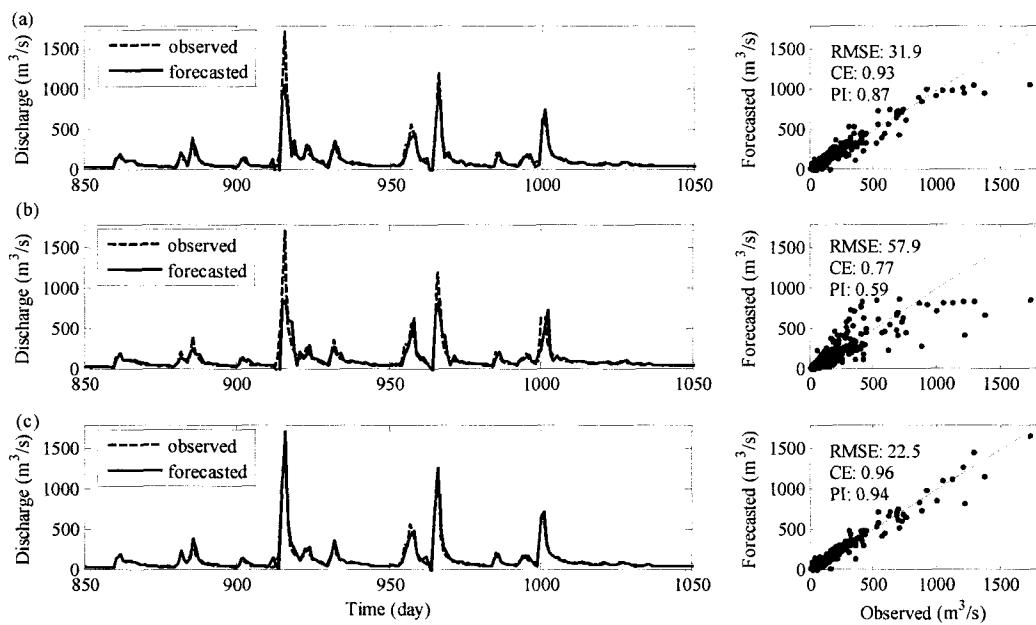


Figure 7.18 Hydrographs and scatter plots of one-step lead forecast using (a) ANN-SSA-A, (b) ANN-WA-A and (c) MANN-SSA-A for the Wuxi data.

The quantitative analysis on the lag effect for daily streamflow data is shown in Figure 7.19. The maximum CCF in each pane is exact at the lag zero, which indicates that SSA-A and WA-A are able to completely eradicate the lag effect (depicted in Figure 7.17).

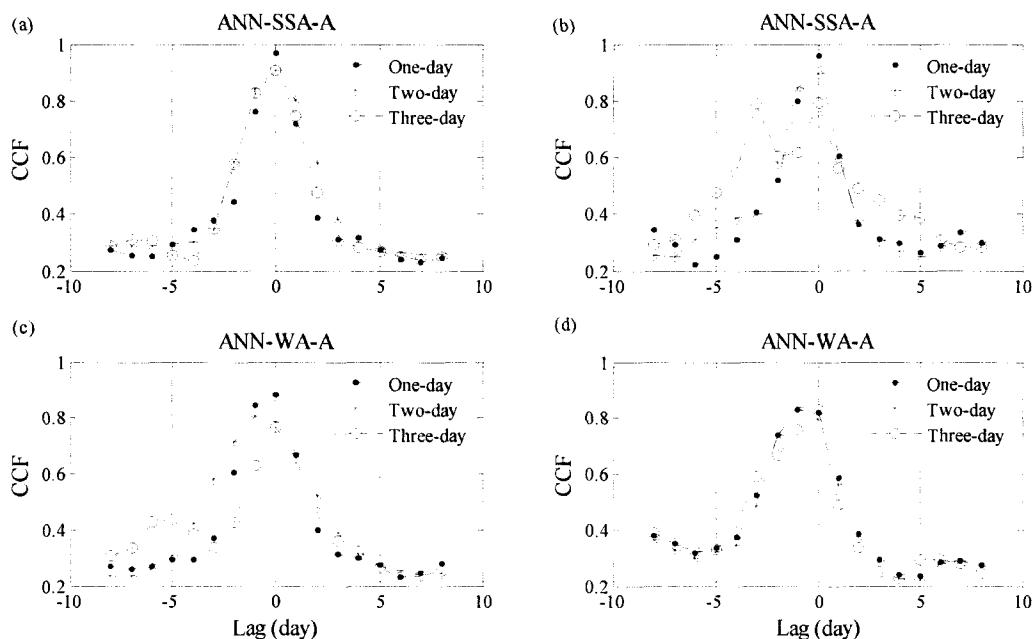


Figure 7.19 CCFs at three forecast horizons using ANN-SSA-A and ANN-WA-A
(The left column for Wuxi, and the right column for Chongyang)

Error analysis is a widely used method to check the suitability of a forecasting model. If a real streamflow process is appropriately reproduced by a proposed model, prediction errors tend to be quasi-normally distributed. In terms of normal distribution, probabilities in the norm plot should be an exact line (as shown in Figure 7.20). Figure 7.20 demonstrates that norm-plots of prediction errors of all four models for Danjiangkou and Wuxi. It is shown that the probabilities of errors are close to the straight line for the Danjiangkou data whereas the probabilities of errors are far from the straight line for the Wuxi data. There is still considerable room for improvement of these models which are used in daily streamflow series although they have good overall performance. The great concern should be put on the improvement of forecasts of peak values.

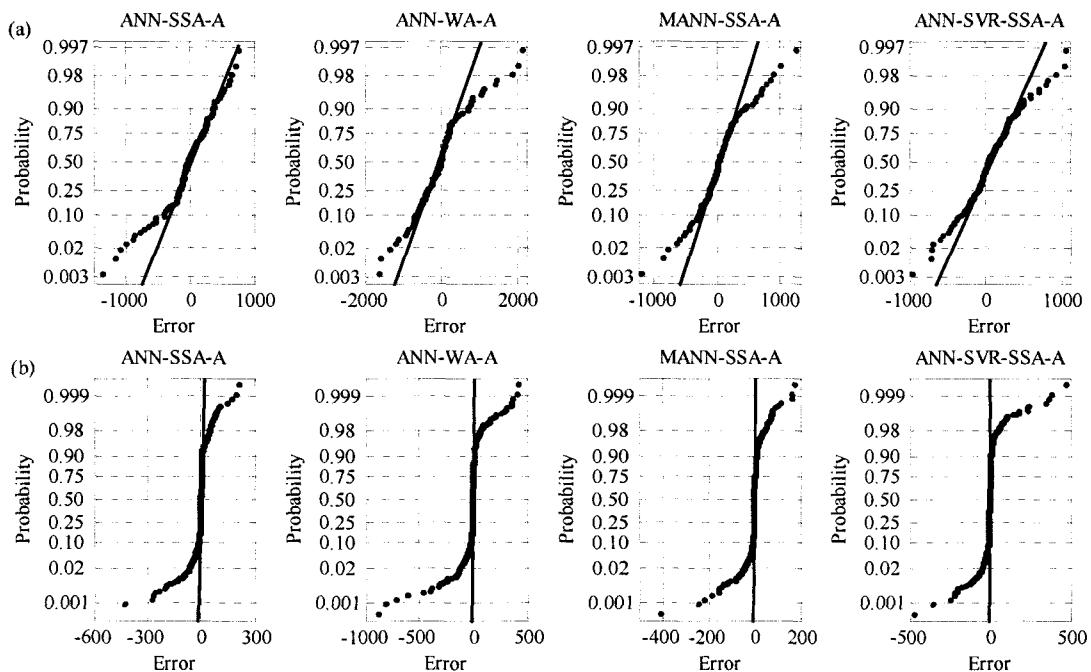


Figure 7.20 Norm plots of prediction errors (predicted-observed) of four models using the flow series of (a) Danjiangkou and (b) Wuxi.

To reveal the effect of SSA-A and WA-A, an analysis of AMI between model input variable and output is presented in Figure 7.21, where ANN is in conjunction with three modes (namely normal mode, SSA-A, and WA-A) for three prediction horizons. As expected, the dependence relation between input variable and output of ANN are remarkably improved by SSA-A or WA-A compared with that in the normal mode, which results in perfect performance of ANN. On the contrary, the individual AMI

seems to be unable to truly reflect the capability of ANN. For example, AMM-SSA-A noticeably outperforms ANN-WA-A at one-step prediction horizon (Table 7.6). However, the AMI of each input in the former is almost smaller than that in the latter. A possible reason is that the performance of ANN is determined by the combination of input variables which interacts through the link of weights. According to the analysis, the requirement of independence of model inputs is in doubt.

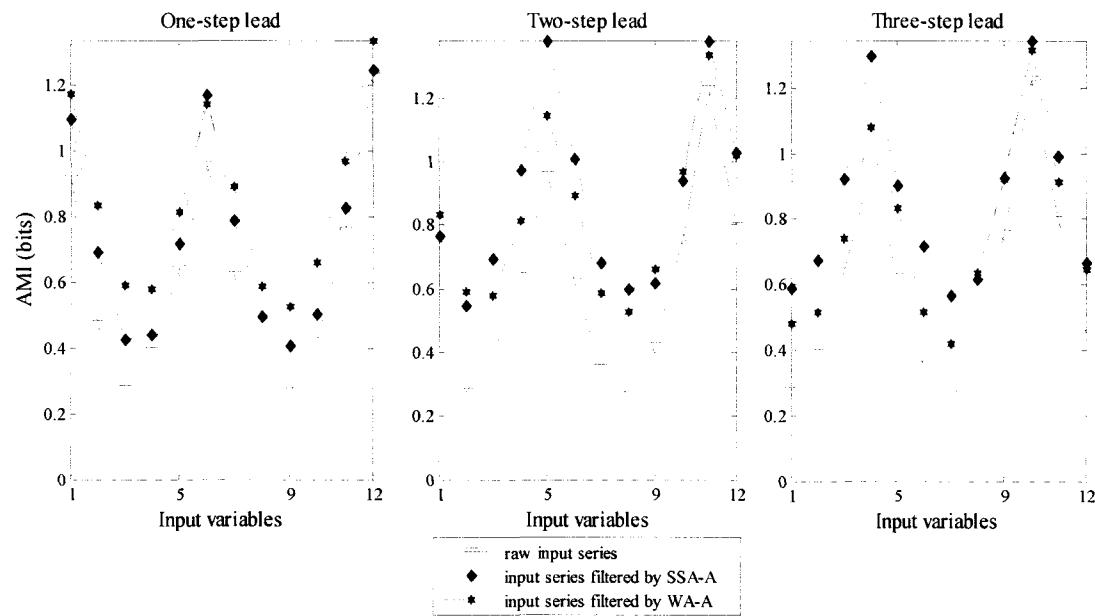


Figure 7.21 AMI between each input variable and output of ANN using the Xiangjiaba data (the number in the abscissa axis denotes such order of inputs as: “1” is the observation at time instant t , “2” is that at time instance $t-1$, and up to “12” which is the observation at time instant $t-11$)

7.4.3 Comparison of schemes A and B

Table 7.7 presents comparison of performance of ANN with SSA and WA in schemes, A and B. In terms of SSA, ANN with scheme A performs better than that with scheme B. In contrast, scheme A does not show any superiority over scheme B in the scenario of WA. Even in one-step prediction of Xiangjiaba, ANN-WA-B is superior to ANN-WA-A. The bad performance of ANN under scheme B may be explained by two potential reasons: one is that scheme B employs the supervised filtering approach for each input variable, which excludes all combinations of RCs (or

DWCs); and the supervised filtering only concerns the improvement of dependence relation between individual input variable and output, but ignores the joint effect of input variables on the model output. To explore the ability of scheme B, a global search method, such as GA, for components filtering of each input variable should be employed. Table 7.7 also shows that the values of PI of ANN-WA-B in two watersheds of Wuxi and Chongyang are small, which implies the existence of the prediction lag effect.

Table 7.7 Comparison of performance of ANN-SSA (or ANN-WA) in schemes A and B

| Watershed | Model | RMSE | | | CE | | | PI | | |
|--------------------|-----------|-------|--------|--------|------|------|------|------|------|------|
| | | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| Xiangjiaba | | | | | | | | | | |
| | ANN-SSA-A | 453.7 | 507.2 | 496.4 | 0.98 | 0.98 | 0.98 | 0.97 | 0.98 | 0.99 |
| | ANN-SSA-B | 735.3 | 754.6 | 835.6 | 0.96 | 0.96 | 0.95 | 0.92 | 0.97 | 0.97 |
| | ANN-WA-A | 937.0 | 996.3 | 1123.1 | 0.93 | 0.92 | 0.90 | 0.86 | 0.94 | 0.95 |
| | ANN-WA-B | 846.3 | 1127.9 | 1139.3 | 0.95 | 0.90 | 0.90 | 0.89 | 0.93 | 0.95 |
| Danjiangkou | | | | | | | | | | |
| | ANN-SSA-A | 329.6 | 312.7 | 328.5 | 0.89 | 0.90 | 0.89 | 0.89 | 0.94 | 0.95 |
| | ANN-SSA-B | 334.8 | 384.5 | 360.3 | 0.89 | 0.86 | 0.87 | 0.88 | 0.90 | 0.94 |
| | ANN-WA-A | 550.9 | 592.9 | 644.7 | 0.70 | 0.66 | 0.59 | 0.69 | 0.77 | 0.81 |
| | ANN-WA-B | 592.0 | 588.9 | 674.9 | 0.65 | 0.66 | 0.55 | 0.64 | 0.78 | 0.79 |
| Wuxi | | | | | | | | | | |
| | ANN-SSA-A | 31.0 | 50.6 | 61.8 | 0.94 | 0.83 | 0.74 | 0.88 | 0.84 | 0.80 |
| | ANN-SSA-B | 35.9 | 58.1 | 52.0 | 0.91 | 0.77 | 0.82 | 0.84 | 0.79 | 0.86 |
| | ANN-WA-A | 55.6 | 76.4 | 78.8 | 0.79 | 0.61 | 0.58 | 0.62 | 0.64 | 0.68 |
| | ANN-WA-B | 68.7 | 77.3 | 89.6 | 0.68 | 0.60 | 0.46 | 0.42 | 0.64 | 0.59 |
| Chongyang | | | | | | | | | | |
| | ANN-SSA-A | 7.9 | 11.2 | 15.7 | 0.91 | 0.81 | 0.63 | 0.86 | 0.83 | 0.72 |
| | ANN-SSA-B | 11.8 | 15.9 | 16.2 | 0.79 | 0.62 | 0.60 | 0.69 | 0.66 | 0.70 |
| | ANN-WA-A | 15.4 | 17.3 | 16.1 | 0.64 | 0.55 | 0.61 | 0.68 | 0.59 | 0.70 |
| | ANN-WA-B | 15.7 | 16.4 | 20.8 | 0.63 | 0.59 | 0.35 | 0.45 | 0.63 | 0.50 |

Figure 7.22 shows an analysis of the lag effect of predictions of ANN in scheme B using daily flow data from Wuxi and Chongyang. ANN-WA-B produces lagged forecasts whereas ANN-SSA-A basically overcomes the lag effect except for the Chongyang watershed at one-step lead.

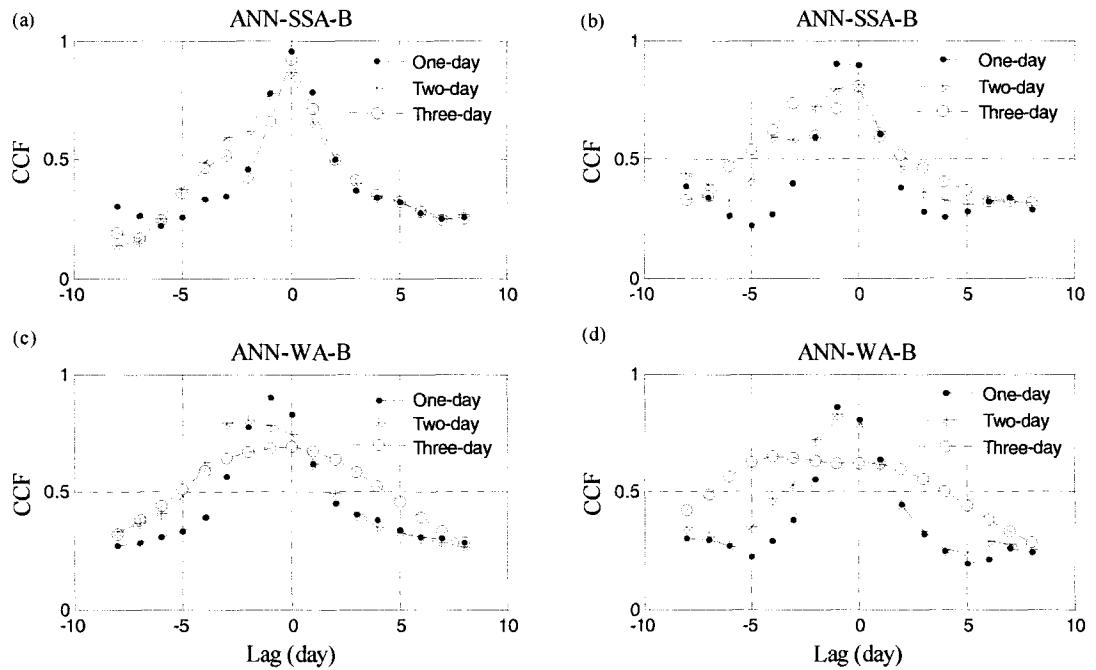


Figure 7.22 CCFs at three forecast horizons using ANN-SSA-B and ANN-WA-B
(The left column for Wuxi, and the right column for Chongyang)

To further compare schemes, A and B, the AMI between input variable and output of each of ANN-SSA-A and ANN-SSA-B are computed and are presented in Figure 7.23 and Figure 7.24 respectively using the Xiangjiaga data and Danjiangkou data. Figure 7.23 shows that the AMI value of each input variable in SSA-A is almost consistently larger than that in SSA-B. However, the situation (see Figure 7.24) is reverse except for few input variables. As shown in Table 7.7, ANN-SSA-A outperforms ANN-SSA-B for each watershed. Therefore, the ambivalent results from Figure 7.23 and Figure 7.24 further prove that the performance of ANN may be governed by combined effects of input variables.

7.4.4 Further research

To conduct a more comprehensive comparison between SSA and WA, a further study needs to focus on the following aspects. First of all, the present study employed the 3 order of Daubechies wavelets as the wavelet function. In general, one function can be viewed as the wavelet function if it has zero mean and be localized in both time and frequency space (Farge, 1992). Evidently, there are a large number of functions to satisfy the admissibility condition. A more appropriate wavelet function may be

found for decomposition of the streamflow series. In the case of SSA, the peak flows cannot be well simulated although ANN or MANN is able to quite accurately make an overall prediction. Finally, it is imperative to develop a global search method for the selection of input components in scheme B.

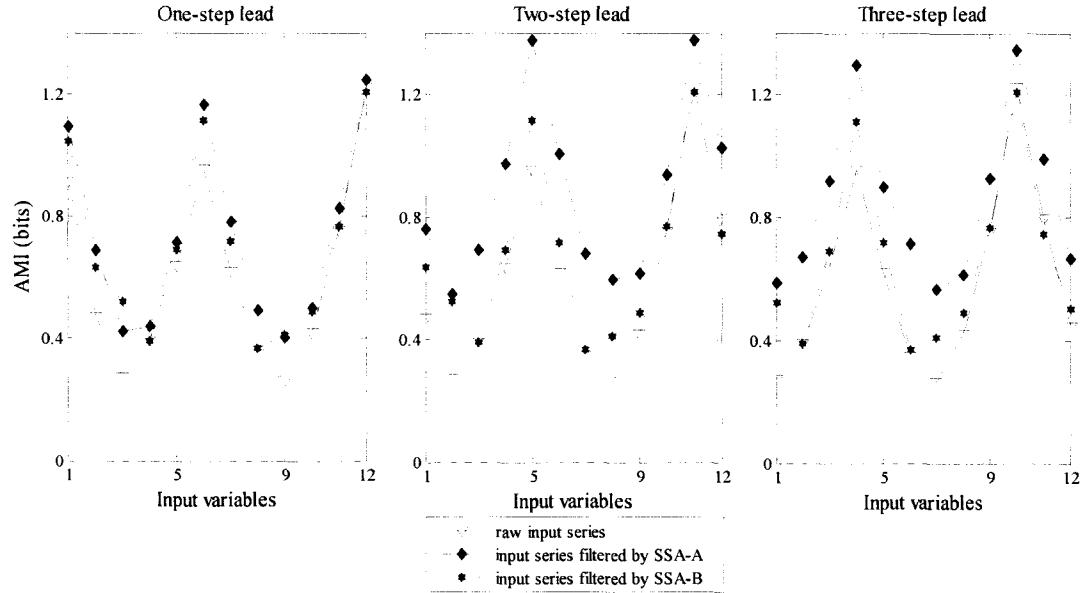


Figure 7.23 AMIs between inputs and output of ANN coupled with three modes at three prediction horizons using the Xiangjiaba data

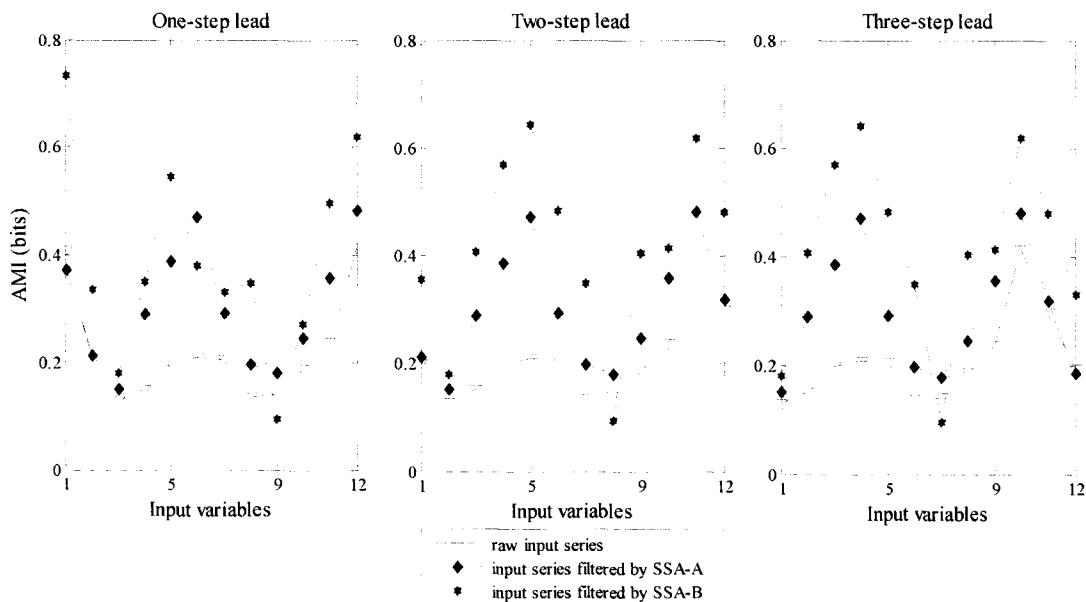


Figure 7.24 AMIs between inputs and output of ANN coupled with three modes at three prediction horizons using the Danjiangkou data

7.5 Summary

This chapter has presented applications of data-driven models in conjunction with SSA and WA to monthly and daily streamflow forecasts. These data-driven models include ANN, MANN, ANN-SVR, and DSBM. Two schemes, A and B, for SSA and WA are compared using the ANN model in the context of monthly and daily flow predictions. The key points can be highlighted as follows:

Similar to the results in Chapter 6, the optimal model input for streamflow forecasting is finally determined by LCA on the basis of comparison amongst all seven data input methods.

In the normal mode, DSBM has been implemented in two ways: local and global. The global DSBM (DSBM-G) outperforms the local DSBM (DSBM-L), and DSBM-G shows similar performance to ANN. Two modular models, MANN and ANN-SVR, are replaceable by each other from the perspective of their close performance. Compared with ANN, the two models are able to noticeably improve the accuracy of flow predictions, in particular for the irregular flow series. However, lagged forecasts widely occur in daily streamflow series.

In scheme A, both SSA and WA can considerably improve the prediction accuracy and completely eradicate the prediction lag effect when they are coupled with ANN, MANN and ANN-SVR. The superiority of modular models over ANN is not significant. The comparison between SSA and WA indicates that SSA is a more effective data preprocessing technique.

Comparison of two schemes, A and B, demonstrates that scheme A is significantly superior to scheme B in the scenario of SSA whereas effects of two schemes are similar in the scenario of WA. It is strongly recommended that a global search method should be used to identify the optimal components for model inputs in scheme B. Finally, proposed forecasting model for streamflow forecasting is ANN coupled with the scheme A SSA because of its relative ease in operation compared with modular models.

8 Rainfall-Runoff Forecasting

The last chapter has presented univariate streamflow prediction in which model inputs consists of past streamflow only. The current chapter attempts to simulate streamflow by the rainfall-runoff (R-R) transformation where potential model inputs may include past streamflow, rainfall, and evaporation. For the purpose of comparison, two watersheds in the last chapter, Wuxi and Chongyang, will be further studied for daily rainfall-runoff predictions using ANN and MANN with LR as benchmark. This chapter primarily concentrates on the following contents:

- (1) To compare LR, ANN and MANN in both the normal mode and the SSA mode;
- (2) To analyze model performance of ANN and ANN-SSA in two types of model input scenarios: using previous flows as the only input variable and using previous flow and rainfall as input variables.

8.1 Introduction

The rainfall-runoff relationship is one of the most complex hydrological phenomena to comprehend, owing to the tremendous spatial and temporal variability of watershed characteristics and precipitation patterns, and to the number of variables involved in the modeling of the physical process (Kumar et al., 2005). Since the rational method for peak of discharge was developed by Mulvany (1850), numerous hydrologic models have been proposed. Based on the description of the governing processes, these models can be classified as either physically-based (knowledge -driven) or system theoretic (data-driven). Physics based models involve a detailed description of various physical processes controlling the hydrologic behavior of a system. However, system theoretic models are instead based primarily on observations (measured data) and seek to characterize the system response from those data using transfer functions. A representative data-driven R-R model can be defined as

$$\hat{Q}_{t+T} = f(Q_{t+1-l_1}, R_{t+1-l_2}, X_{t+1-l_3}) \quad (8.1)$$

where \hat{Q}_{t+T} stands for the predicted flow at time instance $t+T$; T (with $T=1,2$, and 3 for the current study) refers to how far into the future the runoff prediction is desired; Q_{t+1-l_1} is antecedent flow (up to $t+1-l_1$ time steps), R_{t+1-l_2} is antecedent rainfall (up to $t+1-l_2$ time steps) and X_{t+1-l_3} (up to $t+1-l_3$ time steps) represents any other factors contributing to the true flow Q_{t+T} , such as evaporation or temperature. The predictability of future behavior is a consequence of the correct identification of the system transfer function of $f(\bullet)$. Depending on different identification techniques, a plethora of data-driven R-R models have appeared in the literature, including K-NN, ANN, FIS, SVR, and GP. A comprehensive review on them has been given in Chapter 2. According to the literature review, the ANN R-R model has received the greatest attention in the last two decades due to its capability to reproduce the highly nonlinear nature of the relationship between hydrological variables.

The potentiality of ANN in hydrological modeling was reviewed, for example, by the ASCE Task Committee on Application of the ANNs in hydrology (ASCE, 2000b), by Maier and Dandy (2000), and by Dawson and Wilby (2001). The majority of the applications for river flow prediction consist in modeling the R-R transformation, providing input of past flows and the precipitation observations. The majority of studies have proved that ANNs are able to outperform traditional statistical R-R modeling (Hsu et al, 1995; Shamseldin, 1997; Sajikumar and Thandaveswara, 1999; Tokar and Johnson, 1999; Coulibaly et al., 2000; Sudheer et al., 2002) and to offer promising alternatives for conceptual R-R models (Hsu et al, 1995; Tokar and Johnson, 1999; Coulibaly et al., 2000; Dibike and Solomatine, 2001; Birikundavyi et al., 2002; de Vos and Rientjes, 2005; Toth and Brath, 2007). Hsu *et al.* (1995) showed that the ANN model provided a better representation of the rainfall-runoff relationships than the ARMAX time series model or the conceptual SAC-SMA (Sacramento soil moisture accounting) models. Coulibaly et al. (2000) used the early stopping method, to train MLP for real-time reservoir inflow forecasting. Results showed that MLP could provide better model performance compared to benchmarks

from the classic autoregressive model coupled with a Kalman filter (ARMAX-KF) and a conceptual model (PREVIS). Birikundavyi et al. (2002) investigated the ANN models for daily streamflow forecasting and also showed that ANNs outperformed PREVIS and ARMAX-KF. Toth and Brath (2007) investigated the impact of the amount of the training data on model performance using ANN and a conceptual model (ADM). ANN was proved to be an excellent tool for the R-R simulation of continuous periods, provided that an extensive set of hydro-meteorological data was available for calibration purposes. However, compared with ANN, ADM might allow a significant forecasting improvement when focusing on the prediction of flood events and especially in case of a limited availability of the training data.

Improvement of model performance is a long-term topic of interest by researchers when ANN is used to simulate the R-R relationship. It is well known that the ANN model is data dependent and has a flexible structure, which leaves huge room for the improvement of ANN in the context of R-R forecasting. The ANN model is highly sensitive to the studied data, which means that the structure of ANN is totally different with the change of the training data. Besides, the training algorithms, model configuration, and data preprocessing techniques also impose wide influences on the model performance. Hsu et al. (1995) found that the ANN models underestimated low flows and overestimated medium flows when they were used to simulate the R-R relationship. In the discussion part , Hsu et al. (1995) mentioned that this might have been due to the models not being able to capture the nonlinearity in the rainfall-runoff process and suggested that there was still room for improvement in applying different algorithms, such as stochastic global optimization and genetic algorithms, to reach near global solutions, and achieve better model performances. A more effective and efficient ANN R-R model was developed by Jain and Srinivasulu (2004) where ANN was trained by using real-coded GAs. Results showed that the proposed approach can significantly improve the estimation accuracy of the low-magnitude flows.

On the other hand, Zhang and Govindaraju (2000) recently pointed out that the rainfall-runoff mapping in a watershed can be fragmented or discontinuous having significant variations over the input space because of the functional relationships

between rainfall and runoff being quite different for low, medium, and high magnitudes of streamflow. They found a single ANN to be rigid in nature and not suitable in capturing a fragmented input-output mapping. In order to overcome this problem they designed a modular neural network (MANN) consisting of three different ANN models for low-, medium-, and high-magnitude flows. Inspired by this study, many modular (or hybrid) models have been developed for R-R simulation. Solomatine and Xue (2004) applied an approach where separate ANN and M5 model-tree basin models were built for various hydrological regimes (identified on the basis of hydrological domain knowledge). Jain and Srinivasulu (2006) also applied decomposition of the flow hydrograph by a certain threshold value and then built separate ANNs for low and high flow regimes. Corzo and Solomatine (2007) investigated three modular ANNs for simulating two decomposed flow regimes, base flow and exceed flow, depending on three different partitioning schemes: automatic classification based on clustering, temporal segmentation of the hydrograph based on an adapted baseflow separation technique, and an optimized baseflow separation filter. The modular models were shown to be more accurate than the global ANN model. The best modular model was the one using the optimized baseflow filtering equation. Obviously, all studies demonstrated that modular models generally made higher accuracy of prediction than global models built to represent all possible regimes of the modeled system. Hence, MANN continues to be examined in the present study.

When a rainfall or streamflow time series is viewed as combination of quasi-periodic signals contaminated by noises to some extent, a cleaner time series can be filtered by appropriate data preprocessing techniques such as SSA or WA. Obviously, the predictability of a system can be improved by forecasting the important oscillations (periodic components) in time series taken from the system. Referring to Chapters 6 and 7, SSA and WA have been widely used in the forecasting of rainfall or streamflow series alone. However, fewer studies concurrently employ filtered rainfall and streamflow data as inputs of an R-R model. In accordance with the results from chapters 6 and 7, the ANN R-R model with the filtered inputs could be expected to achieve better performance than that with the raw inputs. Therefore, one of main purposes in this study is to propose an ANN R-R model coupled with SSA in

comparison with LR-SSA and MANN-SSA. For the purpose of comparison, two river basins in Chapter 7, Wuxi and Luishui, are examined in the R-R forecasting context.

This chapter is structured in the following manner. Followed by Introduction, two river basins are described in Section 8.2. Section 8.3 presents modeling methods and their applications to two watersheds. The optimal model is identified and the implementation of SSA is described. In Section 8.4, main results are shown along with necessary discussions. Section 8.5 summarizes main conclusions in this chapter.

8.2 Case Studies

Two river basins, Daning (Figure 6.1) and Lushui (Figure 7.1), are considered as case studies. The detailed hydrogeological description of two basins can be found in Chapters 6 and 7. The collected data included streamflow, rainfall, and evaporation. The data period for Daning spans 20 years from January 1, 1988 to December 31, 2007. A 4-year data series from January 1, 2004 to December 31, 2007 is used for analysis of Lushui. In the Daning watershed, the outlet is at the Wuxi station, so it is hereafter referred to as “Wuxi”. Likewise, “Chongyang” denotes the second case study since it is the measure point for streamflow. Figure 8.1 demonstrates rainfall and runoff time series of two basins. The data represents various types of hydrological conditions, and flow range from low to very high. However, it is worth noting that the present experiments are not for a real-time forecasting since the time resolution of the available data is day which is the same order as the reaction time of the catchment.

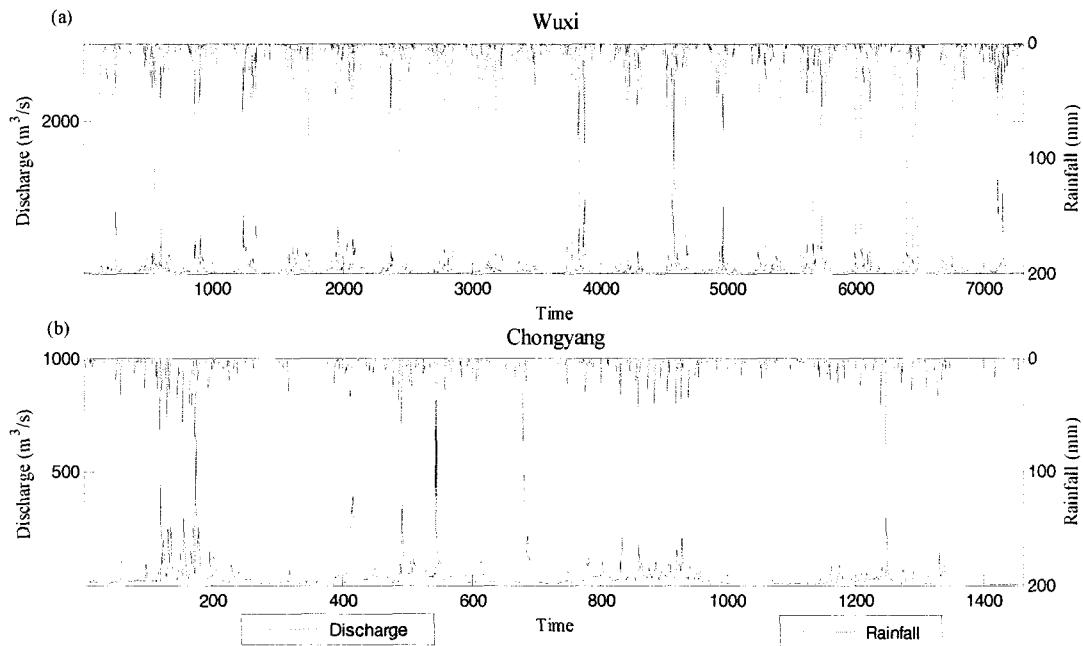


Figure 8.1 Daily rainfall-runoff time series: (a) Wuxi and (b) Chongyang

Each forecasting model is a lumped type, namely, the watershed is considered as a whole, the input rainfall being the mean areal precipitation over the watershed by Thiessen polygon method and the output being the discharge measured at the outlet. The entire input-output dataset in each watershed is partitioned into three data subsets as training set, cross-validation set and testing set: the first half of the entire data as training set and the first half of the remaining data as cross-validation set and the other half as testing set. The training set serves the model training and the testing set was used to evaluate the performances of models. The cross-validation set had dual functions: one was to implement an early stopping approach so as to avoid overfitting of the training data, and another is to select some best predictions from a large number of ANN's runs. Ten best predictions are selected from total twenty ANN's runs in the present study. Table 8.1 presents statistical information on rainfall and streamflow data. Obviously, the training data cannot fully include the cross-validation and testing data in terms of Wuxi. It is recommended that all data be scaled to the interval [-0.9, 0.9] or similar intervals instead of [-1, 1] when ANN employs the hyperbolic tangent sigmoid functions as transfer functions in the hidden layer and output layer.

Table 8.1 Statistical information on rainfall and streamflow data

| Watershed and datasets | Statistical parameters | | | | | | Watershed area and data period | | | |
|--------------------------------|------------------------|-------|-------|-------|------------|------------|--------------------------------|------------------------|--|--|
| | μ | S_x | C_v | C_s | X_{\min} | X_{\max} | | | | |
| Wuxi | | | | | | | | | | |
| <i>Rainfall(mm)</i> | | | | | | | | | | |
| Original data | 3.7 | 10.1 | 0.36 | 5.68 | 0 | 154 | Area: | | | |
| Training | 3.4 | 8.9 | 0.39 | 4.96 | 0 | 102 | | 2 000 km ² | | |
| Cross-validation | 3.8 | 10.9 | 0.35 | 6.27 | 0 | 147 | Data period: | | | |
| Testing | 4.0 | 11.6 | 0.35 | 5.46 | 0 | 154 | | Jan., 1988- Dec., 2007 | | |
| <i>runoff(m³/s)</i> | | | | | | | | | | |
| Original data | 61.9 | 112.6 | 0.55 | 7.20 | 6 | 2230 | | | | |
| Training | 60.6 | 95.6 | 0.63 | 5.90 | 8 | 1530 | | | | |
| Cross-validation | 60.7 | 132.2 | 0.46 | 8.35 | 6 | 2230 | | | | |
| Testing | 66.0 | 122.1 | 0.54 | 6.30 | 10 | 1730 | | | | |
| Chongyang | | | | | | | | | | |
| <i>Rainfall(mm)</i> | | | | | | | | | | |
| Original data | 3.1 | 8.5 | 0.4 | 5.7 | 0.0 | 122 | Area: | | | |
| Training | 3.5 | 9.8 | 0.4 | 5.7 | 0.0 | 122 | | 1 700 km ² | | |
| Cross-validation | 2.9 | 7.0 | 0.4 | 3.9 | 0.0 | 48 | Data period: | | | |
| Testing | 2.6 | 7.0 | 0.4 | 5.6 | 0.0 | 78 | | Jan., 2004- Dec., 2007 | | |
| <i>runoff(m³/s)</i> | | | | | | | | | | |
| Original data | 39.1 | 54.8 | 0.7 | 6.4 | 2.1 | 881 | | | | |
| Training | 48.1 | 70.1 | 0.7 | 5.5 | 6.9 | 881 | | | | |
| Cross-validation | 35.6 | 33.7 | 1.1 | 2.3 | 4.4 | 226 | | | | |
| Testing | 24.5 | 25.7 | 1.0 | 5.1 | 2.1 | 310 | | | | |

8.3 Applications of Models

8.3.1 Implementation framework of R-R forecasting

Two case studies are analyzed by three models, LR, ANN, and MANN, in two modes: with/without SSA. Figure 8.2 illustrates the framework for the R-R forecasting. Five acronyms in the box “methods for model inputs” represent different methods and have been explained in Chapter 6.

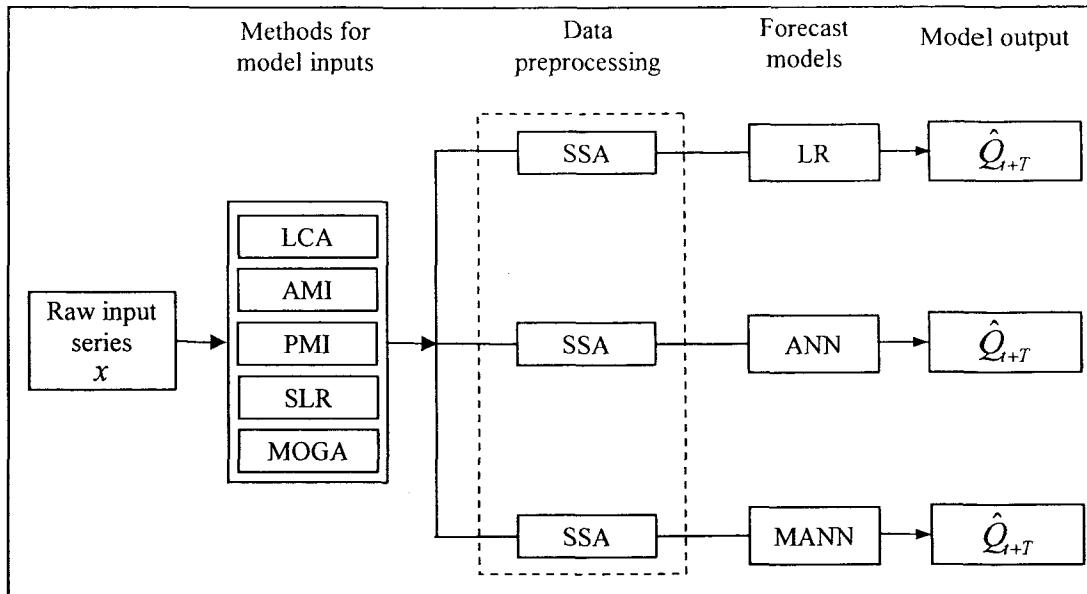


Figure 8.2 Implementation framework of rainfall-runoff forecasting models
with/without data preprocessing

8.3.2 Evaluation of model performances

Three measures used in Chapter 6, CE, RMSE, and PI, are adopted for the present study as follows:

$$CE = 1 - \sum_{i=1}^n (y_i - \hat{y}_i)^2 / \sum_{i=1}^n (y_i - \bar{y})^2 \quad (8.2)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (8.3)$$

$$PI = 1 - \sum_{i=1}^n (y_i - \hat{y}_i)^2 / \sum_{i=1}^n (y_i - y_{i-l})^2 \quad (8.4)$$

In these equations, n is the number of observations, \hat{y}_i stands for forecasted flow, y_i represents observed flow, \bar{y} denotes average observed flow, and y_{i-l} is the flow estimate from a so-called naïve model (or termed persistence model) that basically takes the last flow observation (at time i minus the lead time l) as a prediction. The value of 1 on CE or PI stands for a perfect fit. A small value of PI may imply the occurrence of the lag forecast.

8.3.3 Determination of model inputs

(1) Potential input variables

In the process of determination of model inputs, the first step is to find out appropriate input variables (causal variables) for Eq. (8.1). In general, causal variables in the R-R relationship can be rainfall (precipitation), previous flows, evaporation, temperature etc. Depending on the availability of data, the input variables tend to be various in different studies. The majority of studies employed rainfall and previous flow (or water level) as inputs (Campolo et al., 1999; Liong et al., 2002; Xu and Li, 2002; Sivapragasam et al., 2007) whereas input variables in some studies also included additional factors such as temperature or evaporation (Abrahart et al., 1999; Tokar and Johnson, 1999; Zealand et al, 1999; Zhang and Govindaraju, 2000; Coulibaly et al., 2001; Abebe and Price, 2003; Solomatine and Dulal, 2003; Wilby et al., 2003; Hu et al., 2007; Toth and Brath, 2007; Solomatine and Shrestha, 2009).

The necessity of requirement of previous flows in model inputs was widely recognized by researchers (Campolo et al., 1999; de Vos and Rientjes, 2005). Campolo et al. (1999) made use of distributed rainfall data, observed at different raingauge stations for the prediction of water levels at the catchment outlet. Poor forecasted results were achieved when only water levels were used as input. However, the accuracies of predictions were improved when rainfall and previous water levels were included in inputs. de Vos and Rientjes (2005) employed different model inputs as hydrological state representation of ANN. Results also showed that the ANN model with rainfall input variable only had the worst performance compared to those whose input variables consisting of rainfall, flow and/or other states.

In the meantime, some studies pointed out that evaporation (or temperature) as input variable seemed to be unnecessary (Abrahart et al., 2001; Anctil et al., 2004; Toth and Brath, 2007). Anctil et al. (2004) found that potential evapotranspiration failed to improve the MLP performance when it was introduced into the initial model inputs consisting of rainfall and streamflow for R-R modeling. Results from Toth and Brath (2007) also indicated that the inclusion of potential evapotranspiration values in

inputs did not improve the forecasting results, but gave rise to a slight deterioration in comparison with the use of precipitation data alone. That result might be explained by the fact that the addition of evapotranspiration (or temperature measures) input nodes increased the network complexity, and therefore the risk of overfitting. In the present experiments, analyses of LCA, AMI, and PMI between evaporation and streamflow indicate that evaporation appears to be negligible since the dependence relation is not significant. Therefore, rainfall and streamflow are identified as final input variables.

(2) Selection of model inputs

Having chosen appropriate input variables, the next step is determination of appropriate lags for each of them so as to form model inputs. ANN, equipped with the L-M training algorithm and hyperbolic tangent sigmoid transfer functions, is used as the benchmark model to examine five input methods. FNN and CI are not considered since they are suitable exclusively for the analysis of univariate time series.

The number l_1 of lags of flow was initially set at the value of 5 for Wuxi and 4 for Chongyang according to the results from Figure 7.5. The number l_2 of lags of rainfall is generally determined according to the time of concentration of the watershed. The time of concentration used herein was estimated between the center of hyetograph and the peak flow. The average time of concentration was approximately 1 day for Wuxi and Chongyang. To take account of the delay between rainfall and runoff, the value of l_2 is originally set to 5 for both Wuxi and Chongyang. Table 8.2 presents the results of ANN with different model inputs determined by LCA, AMI, PMI, SLR and MOGA. These results are based on one-step-ahead flow prediction (i.e. \hat{Q}_{t+1} where t represents the present time instance). In terms of RMSE of ANN, there is no salient difference amongst all five methods. However, it is found that the ANN with inputs from LCA outperforms the others in the later SSA scenario. Moreover, LCA can significantly reduce the effort and computational time requirement in developing an ANN model. The LCA method is therefore adopted for the later analysis. Figure 8.3 illustrates CCFs between rainfall

and streamflow for Wuxi and Chongyang. The past five rainfall observations had significant relations (correlation coefficient > 0.2) with the present streamflow. The most significant correlation occurs at the first lag which indicates the time of response of watershed being about 1 day.

Table 8.2 Comparison of methods to determine mode inputs using ANN

| Watershed | Methods | τ | l_1 | l_2 | m | Effective inputs | Identified ANN | RMSE |
|------------------|---------|--------|-------|-------|-----|--------------------------------|----------------|-------|
| Wuxi | | | | | | | | |
| | LCA | 1 | 5 | 5 | 10 | all | (10-8-1) | 41.98 |
| | AMI | 1 | 5 | 5 | 10 | all | (10-8-1) | 41.98 |
| | PMI | 1 | 5 | 5 | 10 | all | (10-8-1) | 41.98 |
| | SLR | 1 | 5 | 5 | 10 | except for Rt-3 | (9-5-1) | 40.54 |
| | MOGA | 1 | 5 | 5 | 10 | Rt,t-1,t-2,t-3,t-4, Qt,t-1,t-4 | (8-6-1) | 43.23 |
| Chongyang | | | | | | | | |
| | LCA | 1 | 5 | 4 | 9 | all | (9-9-1) | 14.43 |
| | AMI | 1 | 5 | 4 | 9 | except for Rt | (8-7-1) | 14.18 |
| | PMI | 1 | 5 | 4 | 9 | except for Rt | (8-7-1) | 14.18 |
| | SLR | 1 | 5 | 4 | 9 | except for Rt-1,t-2,t-4 | (6-9-1) | 13.54 |
| | MOGA | 1 | 5 | 4 | 9 | Rt,t-1,t-2,t-4, Qt,t-2,t-3 | (7-5-1) | 13.57 |

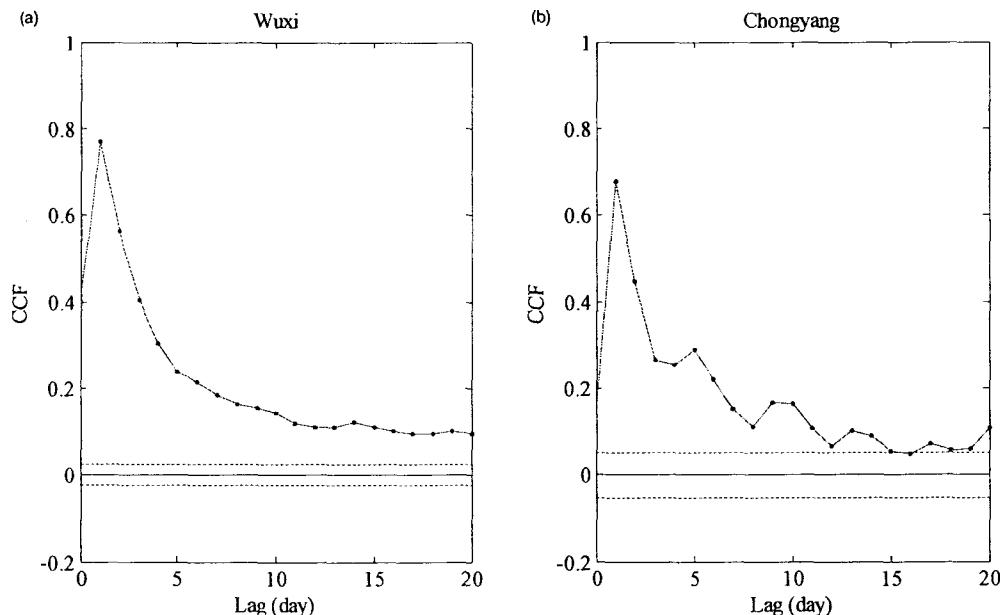


Figure 8.3 CCFs between rainfall and flow series with the 95% confidence bounds (the dashed lines): (a) for Wuxi, and (b) for Chongyang.

8.3.4 Identification of models

The model identification of a forecasting model is to determine the structure by using training data to optimize relevant model parameters once model inputs have already been obtained.

(1) LR

LR can be viewed as a mode-driven model which has a known model structure. Model identification only consists in optimizing the coefficient of each input. The SLR technique is used to concurrently determine the model inputs and the corresponding coefficients. With model inputs already obtained by SLR in Table 8.2, the LR model at one-step lead for Wuxi and Chongyang can expressed respectively as Eq. (8.5),

$$\begin{aligned}\hat{Q}_{t+1} = & -0.019Q_{t-4} + 0.025Q_{t-2} + 0.016Q_{t-1} + 0.469Q_t + 0.046R_{t-4} + \\ & 0.07R_{t-3} + 0.027R_{t-2} + 0.121R_{t-1} + 0.272R_t\end{aligned}\quad (8.5)$$

and Eq. (8.6),

$$\hat{Q}_{t+1} = 0.032Q_{t-3} + 0.526Q_t + 0.099R_{t-3} + 0.053R_{t-2} + 0.037R_{t-1} + 0.454R_t \quad (8.6)$$

(2) ANN and MANN

As a three-layer perceptron is adopted, the identification of the ANN structure is to optimize the number of hidden nodes h in the hidden layer when the model inputs have been determined by LCA and there is a unique model output. The optimal size h of the hidden layer was found by systematically increasing the number of hidden neurons from 1 to 10 until the network performance on the cross-validation set no longer improves significantly. The identified configurations of ANN were 10-8-1 for Wuxi and 9-9-1 for Chongyang, respectively (presented in Table 8.2). The same method was used to identify three local ANNs in MANN. As a consequence, the structures of MANN are 10-4/4/2-1 for Wuxi and 9-3/3/1-1 for Chongyang, respectively.

In order to perform multi-step-ahead predictions, two methods are available: (1) re-using a one-step-ahead prediction as input into the network, after which it predicts the two-step-ahead prediction, and so forth, and (2) by directly having the multi-step-ahead prediction as output. The former and the latter are respectively termed the dynamic model and static model. For simplification, the static model is adopted herein. The static model structure at each prediction horizon is presented in Table 8.3.

Table 8.3 Model structure of static multi-step forecasting

| Watershed | Prediction horizons | Model inputs | | | | | | | | | | Model output | | |
|------------------|---------------------|--------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|--------------|-----------|-----------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1 | 2 | 3 |
| Wuxi | | | | | | | | | | | | | | |
| | 1-day-ahead | Q_t | Q_{t-1} | Q_{t-2} | Q_{t-3} | Q_{t-4} | R_t | R_{t-1} | R_{t-2} | R_{t-3} | R_{t-4} | Q_{t+1} | | |
| | 2-day-ahead | Q_t | Q_{t-1} | Q_{t-2} | Q_{t-3} | Q_{t-4} | R_t | R_{t-1} | R_{t-2} | R_{t-3} | R_{t-4} | | Q_{t+2} | |
| | 3-day-ahead | Q_t | Q_{t-1} | Q_{t-2} | Q_{t-3} | Q_{t-4} | R_t | R_{t-1} | R_{t-2} | R_{t-3} | R_{t-4} | | | Q_{t+3} |
| Chongyang | | | | | | | | | | | | | | |
| | 1-day-ahead | Q_t | Q_{t-1} | Q_{t-2} | Q_{t-3} | R_t | R_{t-1} | R_{t-2} | R_{t-3} | R_{t-4} | | Q_{t+1} | | |
| | 2-day-ahead | Q_t | Q_{t-1} | Q_{t-2} | Q_{t-3} | R_t | R_{t-1} | R_{t-2} | R_{t-3} | R_{t-4} | | | Q_{t+2} | |
| | 3-day-ahead | Q_t | Q_{t-1} | Q_{t-2} | Q_{t-3} | R_t | R_{t-1} | R_{t-2} | R_{t-3} | R_{t-4} | | | | Q_{t+3} |

8.3.5 Decomposition of input series by SSA

The single-channel SSA respectively has been imposed respectively on the rainfall series and the streamflow series. Hence, operational procedures of decompositions on them were the same as those in Chapters 6 and 7. The crucial point relies on selecting appropriate parameter pair (τ, L) . The sensitivity analysis method mentioned in previous two chapters is still adopted herein. A target L can be empirically determined in accordance with the criterion: the singular spectrum under the target L can be distinguished markedly. Figure 8.4 illustrates the sensitivity analysis of singular spectrum on L for rainfall and streamflow series from the two basins of Wuxi and Chongyang. The singular spectrum of both rainfall and flow series in the Wuxi watershed is clearly separate. Clearly, L can be arbitrarily chosen from 3 to 10 in terms of the criterion. To obtain a more robust ANN model, it is recommended that a larger L be taken which results in more combinations of RCs in the process of seeking the optimal model inputs. Thus, the final L is set at the value of 9 for the Wuxi rainfall, 7 for the Wuxi flow, 7 for the Chongyang rainfall and flow. Figure 8.4 highlights the singular spectrum curve associated with the selected L in the dotted line.

Figure 8.5 shows the results of sensitivity analysis of the singular spectrum on the lag time τ using SSA with the chosen L . The singular spectrum can be clearly distinguished at $\tau = 1$. Therefore, the final parameter pair (τ, L) in SSA is set as (1, 9) for the Wuxi rainfall, and (1, 7) for the other three series, respectively. Thus, each rainfall or flow series can be decomposed into RCs with these identified parameter pair.

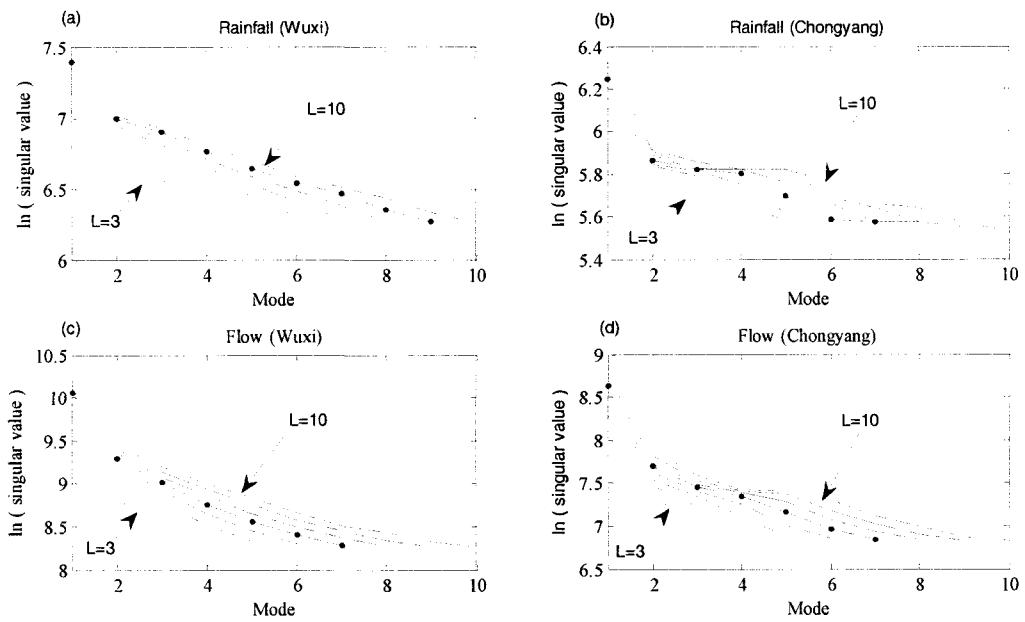


Figure 8.4 Singular Spectrum as a function of lag using varied window lengths L from 3 to 10: (a) and (c) for Wuxi, and (b) and (d) for Chongyang.

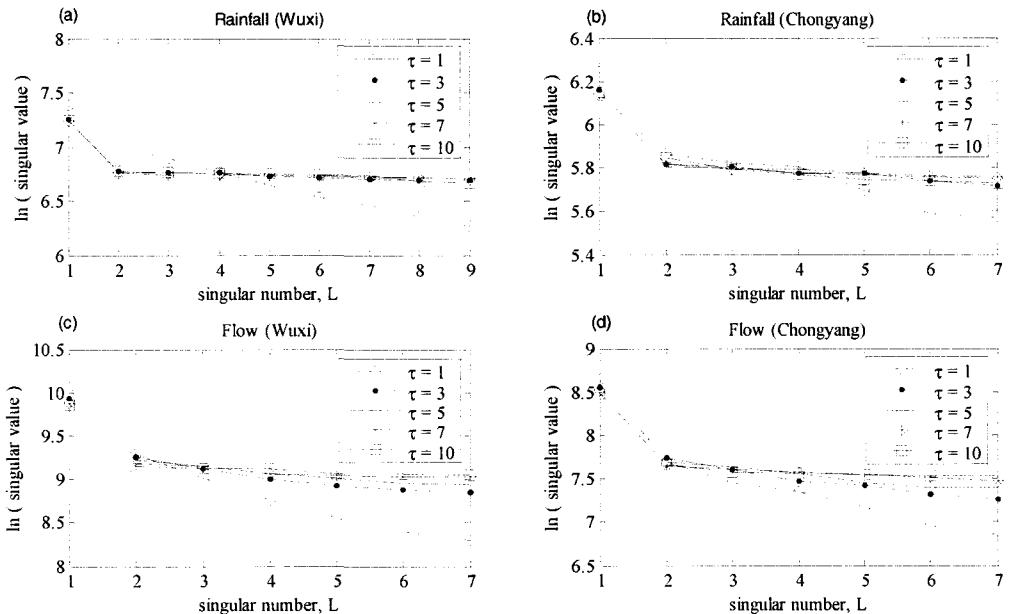


Figure 8.5 Sensitivity analysis of singular Spectrum on various τ : (a) and (c) for Wuxi and (b) and (d) for Chongyang.

8.3.6 Combination of models with SSA

Once the rainfall series and flow series in each watershed are decomposed into RCs, the subsequent task is to generate new model inputs by the unsupervised filtering

method (referring to Chapter 6). Since input variables consist of rainfall and flow, the filtering procedure has to be conducted separately for each variable. Taking the ANN R-R model with SSA (hereafter referred to as ANN-SSA) as an example, two new ANN models need to be established for the purpose of RCs' filtering, one for rainfall input and the other for streamflow input. For the convenience of identification, the ANN model for rainfall input filtering is denoted by ANN-RF, and the ANN model for streamflow input filtering is referred to as ANN-QF. ANN-RF had the same model output as that of the original ANN R-R model and its model input is the same as the rainfall input part of the original ANN R-R model inputs. Likewise, the ANN-QF model input is from the streamflow input part of the original ANN R-R model inputs, and both of them had the same model output variable. Depending on trial and error, ANN-RF and ANN-QF can be identified. For example, ANN-RF is 5-3-1 for Wuxi and 5-4-1 for Chongyang, respectively, and ANN-QF is 5-4-1 for Wuxi and 4-1-1 for Chongyang, respectively. Similarly, LR-RF and LR-QF are also developed for the RCs filter of both rainfall and streamflow series in the context of LR. Table 8.4 presents the RCs filtering results of input variables of rainfall and streamflow for both LR-SSA and ANN-SSA (or MANN-SSA). Two basic conclusions can be drawn from Table 8.4 in the context of SSA: ANN-SSA outperforms LR-SSA with the same model inputs; the model with only streamflow input, either LR-SSA or ANN-SSA, performs better than that with only rainfall input. Therefore, inclusion of flow in model inputs proves to be imperative in R-R forecasting.

Table 8.4 Optimal p RCs for rainfall and streamflow variables at various forecasting horizons

| Filter model | Prediction horizons | Wuxi | | | | | Chongyang | | | | |
|---------------|---------------------|-----------------|-----|---|----------------|-------|-----------------|---|---|---|-------|
| | | Optimal p RCs | | | | RMSE | Optimal p RCs | | | | RMSE |
| LR-RF | 1 | all | RCs | | | 57.13 | 1 | 3 | | | 25.88 |
| | 2 | 1 | 2 | 3 | 5 ^a | 58.37 | 1 | 2 | 6 | | 25.81 |
| | 3 | 1 | 2 | 3 | | 74.24 | 1 | 2 | 7 | | 25.49 |
| LR-QF | 1 | 1 | 2 | 3 | | 35.83 | 1 | 2 | 3 | | 8.92 |
| | 2 | 1 | 2 | | | 55.94 | 1 | 2 | | | 13.41 |
| | 3 | 1 | | | | 67.60 | 1 | | | | 16.60 |
| ANN-RF | 1 | 1 | 3 | 4 | 6 | 7 | 1 | 3 | 5 | 7 | 18.45 |
| | 2 | 1 | 2 | 3 | 4 | 5 | 1 | 3 | | | 19.11 |
| | 3 | 1 | 2 | 3 | 4 | | 1 | 2 | | | 21.72 |
| ANN-QF | 1 | 1 | 2 | 3 | 4 | | 31.49 | 1 | 2 | 3 | 11.67 |
| | 2 | 1 | 2 | 7 | | | 45.39 | 1 | 2 | | 14.97 |
| | 3 | 3 | 7 | | | 53.55 | 1 | | | | 17.26 |

Note: ^a numbers of "1, 2, 3, 5" stand for RC, RC2, RC4, and RC5, and RC1 is associated with the maximum eigenvalue, RC2 corresponds to the second largest eigenvalue, etc.

8.4 Results and Discussions

Results of R-R prediction are presented respectively according to the normal mode and SSA mode. In each mode, three models of LR, ANN, and MANN are compared by three model performance indices. In the SSA mode, three models are referred to as LR-SSA, ANN-SSA, and MANN-SSA.

8.4.1 Forecasts in normal mode

As seen from Table 8.5, all models except for LR for Chongyang have made one-step-ahead predictions with good accuracy. This indicates that causal variables of model output have been accurately selected for this prediction horizon. The performance of each model deteriorates abruptly with the increase of prediction horizons, which may indicate the adoption of inappropriate model inputs. Basically, it is intuitive that a poor prediction on the testing set may result from the lack of similar patterns between the training set and testing set. Conversely, an excellent prediction probably means that there are a large number of similar patterns between them. For example, all models perform better using the Wuxi data than using the Chongyang data since the former has a large size training data (ten years) which allows models to be appropriately trained. A conclusion can also be drawn that ANN (or MANN) tends to be superior to LR if the mapping relation is appropriately identified. The superiority of MANN over ANN seems to be sensitive to the studied data.

Table 8.5 R-R model performances at three forecasting horizons under the normal mode

| Watershed | Model | RMSE | | | CE | | | PI | | |
|------------------|-------|-------|-------|--------|------|-------|-------|------|------|------|
| | | 1* | 2* | 3* | 1 | 2 | 3 | 1 | 2 | 3 |
| Wuxi | | | | | | | | | | |
| | LR | 49.40 | 89.40 | 108.90 | 0.84 | 0.46 | 0.21 | 0.70 | 0.51 | 0.39 |
| | ANN | 43.97 | 87.32 | 104.94 | 0.87 | 0.49 | 0.26 | 0.76 | 0.54 | 0.43 |
| | MANN | 40.44 | 71.87 | 86.54 | 0.89 | 0.66 | 0.50 | 0.80 | 0.69 | 0.61 |
| Chongyang | | | | | | | | | | |
| | LR | 19.18 | 22.74 | 25.53 | 0.44 | 0.22 | 0.01 | 0.17 | 0.29 | 0.24 |
| | ANN | 12.90 | 25.80 | 27.81 | 0.75 | 0.10 | -0.15 | 0.63 | 0.10 | 0.13 |
| | MANN | 13.27 | 26.86 | 23.96 | 0.74 | -0.07 | 0.14 | 0.61 | 0.03 | 0.35 |

* Numbers of “1, 2, and 3” denote one-, two-, and three-step-ahead forecasts

Figure 8.6 and Figure 8.7 illustrate representative details of hydrographs and scatter plots of one-step-ahead forecast using three forecast models for Wuxi and Chongyang,

respectively. The scatter plot from the LR model with high spread at the magnitude of low flows indicates poor forecasts of low flows compared with scatter plots from both ANN and MANN. The principal drawback of ANN and MANN is that they fairly underestimate or overestimate peak flows but reproduced low flows appropriately.

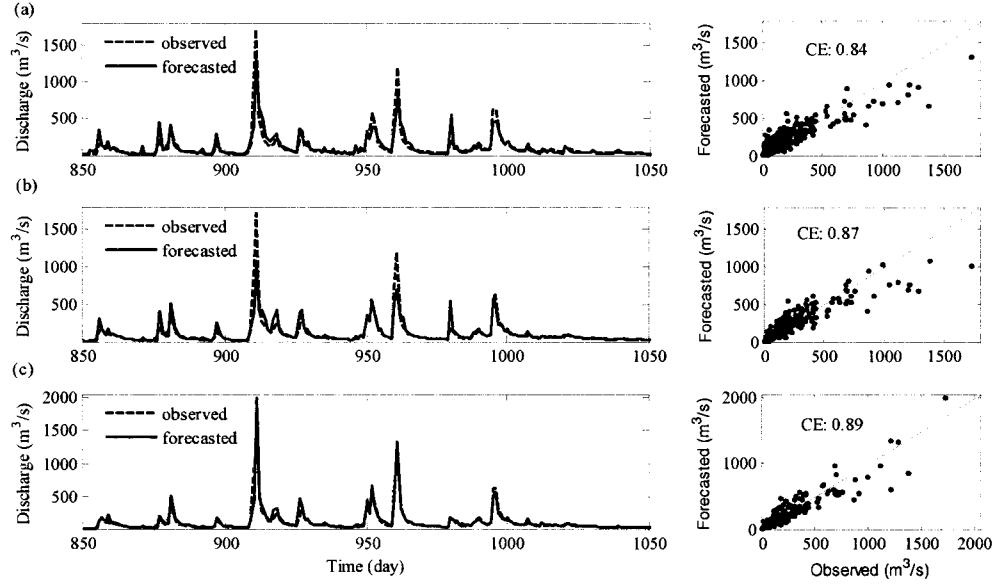


Figure 8.6 Hydrographs (representative details) and scatter plots of one-step-ahead prediction for Wuxi: (a) LR, (b) ANN, and (c) MANN

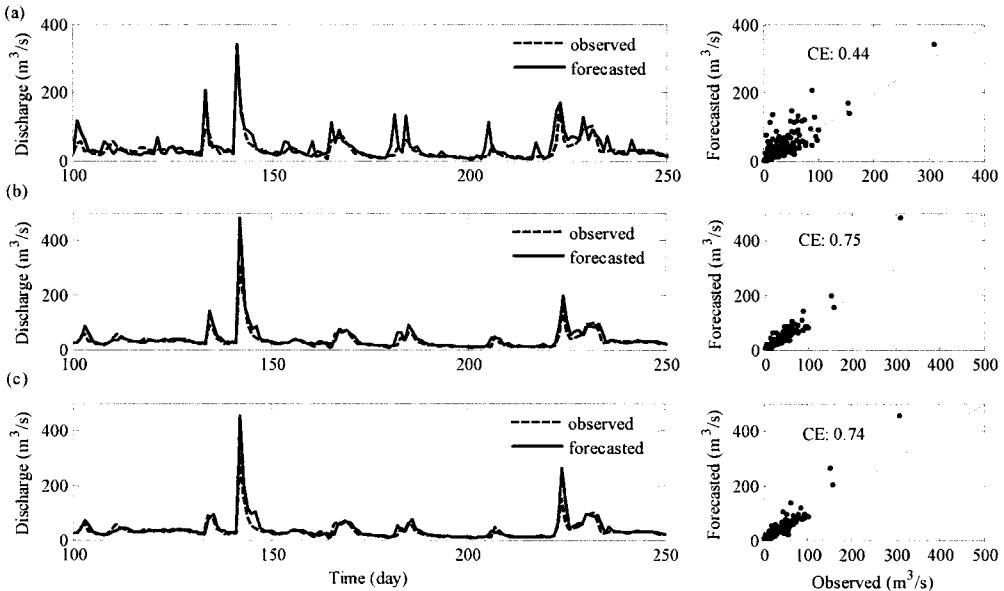


Figure 8.7 Hydrographs (representative details) and scatter plots of one-step-ahead prediction for Chongyang: (a) LR, (b) ANN, and (c) MANN.

Table 8.6 presents comparison of streamflow forecasts using ANN and MANN with two types of inputs: past flow as only input variable, and previous rainfall and flow

as input variables. The forecasts using flows itself as model inputs are taken from Chapter 7. It can be observed that, for the study case of Wuxi, the inclusion of rainfall in input results in the improvement of model performance irrespective of ANN and MANN. However, the degree of the improvement mitigates with the increase of forecast leads. This may indicate that the influence of rainfall on runoff gradually weakens with the increase of prediction horizons. A reverse result was found by Sivapragasan et al. (2007) in which the influence of rainfall on runoff (the time resolution of the data is fortnightly) gradually increased with increasing prediction lead. Employing the data with an hourly time resolution, Toth and Brath (2007) investigated the performance of ANN in two types of inputs. They found that ANN with the inclusion of rainfall in input outperformed ANN with only flow as input at all prediction leads from 1 hour up to 12 hours. Actually, whether or not rainfall is introduced to input heavily relies on the characteristic of the studied watershed. In general, inclusion of rainfall in input is imperative if the prediction lead is less than the average time of concentration. The time of concentration can be roughly identified by the AMI (or CCF) analysis between available rainfall and flow data, and it approximately equals the maximum AMI (or CCF). As shown in Figure 8.3, the time of concentration in each basin is around one day, either more or less. If the time resolution of data is hourly-based, the time of concentration can be approximated to an hour. Therefore, the inclusion of rainfall in input has led to a noticeable improvement of accuracy of one-day-ahead prediction. In this regard, a more detailed analysis will be addressed in the section of discussions.

Table 8.6 Performances of ANN and MANN in two types of input variables

| Watershed | Input variables | Model | RMSE | | | CE | | | PI | | | |
|------------------|----------------------|-------|-------|--------|---|------|-------|-------|------|------|------|--|
| | | | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | |
| Wuxi | | | | | | | | | | | | |
| | <i>Rainfall+Flow</i> | | | | | | | | | | | |
| | ANN | 43.97 | 87.32 | 104.94 | | 0.87 | 0.49 | 0.26 | 0.76 | 0.54 | 0.43 | |
| | MANN | 40.44 | 71.87 | 86.54 | | 0.89 | 0.66 | 0.50 | 0.80 | 0.69 | 0.61 | |
| | <i>Flow</i> | | | | | | | | | | | |
| | ANN | 81.3 | 104.6 | 111.5 | | 0.56 | 0.27 | 0.17 | 0.19 | 0.33 | 0.36 | |
| | MANN | 75.7 | 93.7 | 97.1 | | 0.62 | 0.41 | 0.37 | 0.30 | 0.46 | 0.51 | |
| Chongyang | | | | | | | | | | | | |
| | <i>Rainfall+Flow</i> | | | | | | | | | | | |
| | ANN | 12.90 | 25.80 | 27.81 | | 0.75 | 0.10 | -0.15 | 0.63 | 0.10 | 0.13 | |
| | MANN | 13.27 | 26.86 | 23.96 | | 0.74 | -0.07 | 0.14 | 0.61 | 0.03 | 0.35 | |
| | <i>Flow</i> | | | | | | | | | | | |
| | ANN | 20.3 | 26.1 | 27.8 | | 0.38 | -0.04 | -0.18 | 0.08 | 0.06 | 0.10 | |
| | MANN | 17.8 | 22.3 | 23.4 | | 0.52 | 0.24 | 0.17 | 0.29 | 0.31 | 0.36 | |

The hydrograph of one-step-ahead prediction is presented in Figure 8.8. The ANN

model with only flow input makes the lagged predictions whereas the ANN model with rainfall and flow as inputs eliminates the lag effect. However, with the increase of prediction leads, each of two types of ANN yields the prediction lag effect as shown in Figure 8.9, which indicates the effect of rainfall on model output mitigated markedly.

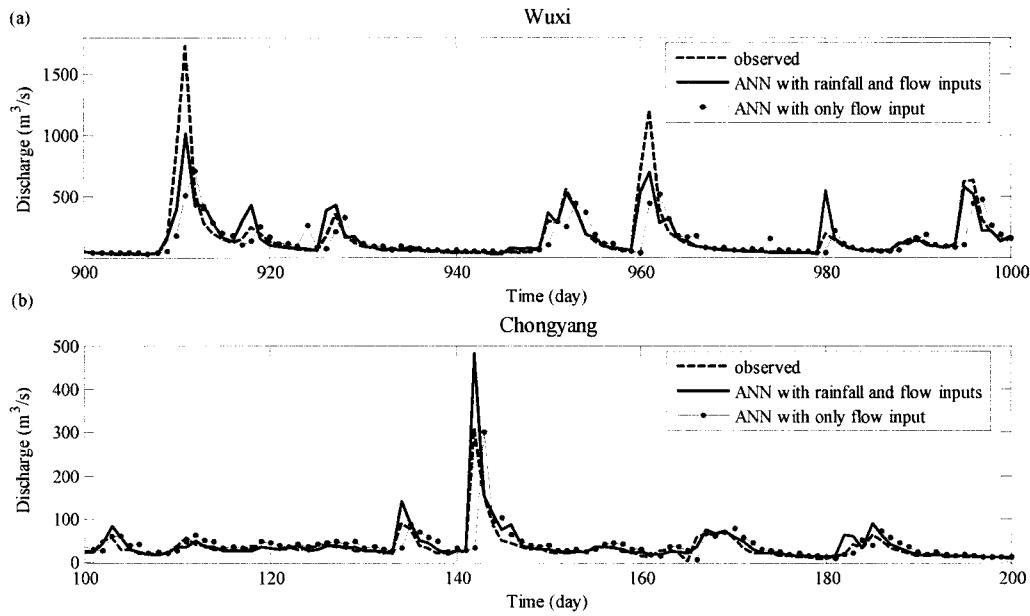


Figure 8.8 Hydrographs for one-step-ahead prediction using ANN with two types of inputs: (a) Wuxi, and (b) Chongyang.

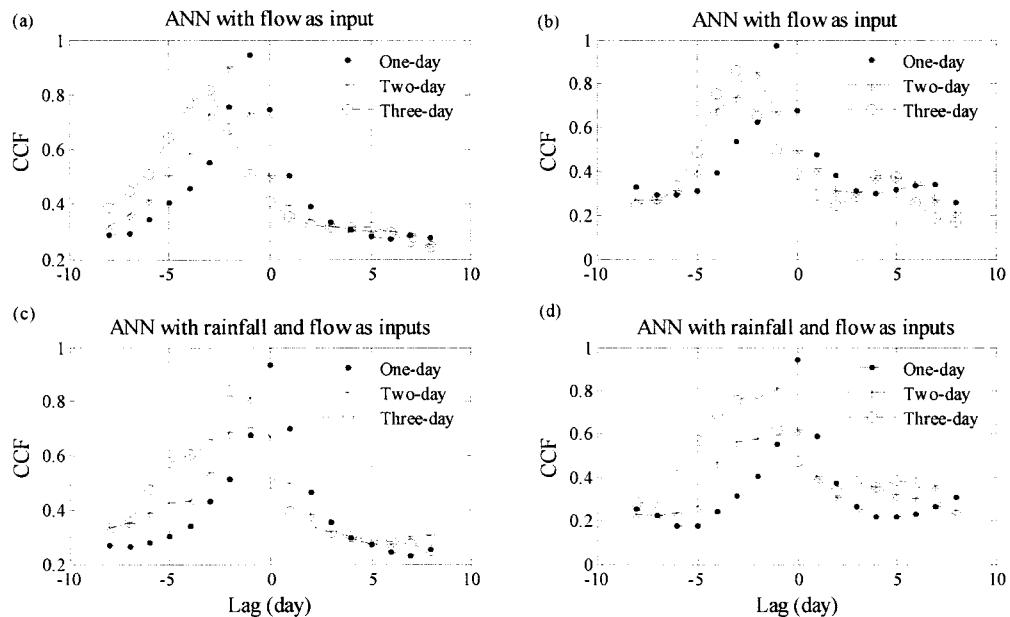


Figure 8.9 Lag analysis of observation and forecasts of ANN with two types of inputs: (a) and (c) for Wuxi, and (b) and (d) for Chongyang.

8.4.2 Forecasts in SSA mode

Table 8.7 presents the results of R-R predictions for Wuxi and Chongyang using three forecasting models coupled with SSA. Compared with the results of Table 8.5, the SSA technique brings about a significant improvement of model performance at all three prediction horizons. Models of ANN and MANN outperform the LR model, but the MANN model exhibits no superiority over the ANN model.

Table 8.7 Performances of R-R models in the SSA mode

| Watershed | Model | RMSE | | | CE | | | PI | | |
|------------------|----------|-------|-------|-------|------|------|------|------|------|------|
| | | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| Wuxi | | | | | | | | | | |
| | LR-SSA | 29.02 | 44.42 | 58.34 | 0.94 | 0.87 | 0.77 | 0.90 | 0.88 | 0.82 |
| | ANN-SSA | 25.40 | 27.10 | 33.96 | 0.96 | 0.95 | 0.92 | 0.92 | 0.96 | 0.94 |
| | MANN-SSA | 25.08 | 26.87 | 34.05 | 0.96 | 0.95 | 0.92 | 0.92 | 0.96 | 0.94 |
| Chongyang | | | | | | | | | | |
| | LR-SSA | 9.19 | 13.53 | 14.61 | 0.87 | 0.72 | 0.68 | 0.81 | 0.75 | 0.75 |
| | ANN-SSA | 6.22 | 7.08 | 11.12 | 0.94 | 0.93 | 0.82 | 0.91 | 0.93 | 0.86 |
| | MANN-SSA | 6.42 | 8.13 | 13.14 | 0.94 | 0.90 | 0.74 | 0.91 | 0.91 | 0.80 |

The representative details of hydrograph and scatter plots of one-step-ahead prediction for Wuxi and Chongyang are shown in Figure 8.10 and Figure 8.11 respectively. These results show that three models with SSA are able to produce good predictions because the predicted hydrograph perfectly reproduces the actual hydrograph and the scatter plots are close to the exact line with a rather low spread. It can be observed from the hydrograph that the LR-SSA model produces some negative forecasts for the low flows and ANN-SSA and MANN-SSA occasionally make negative forecasts at the low-flow points. The peak values are still overestimated or underestimated although each model with SSA exhibits excellent overall performances.

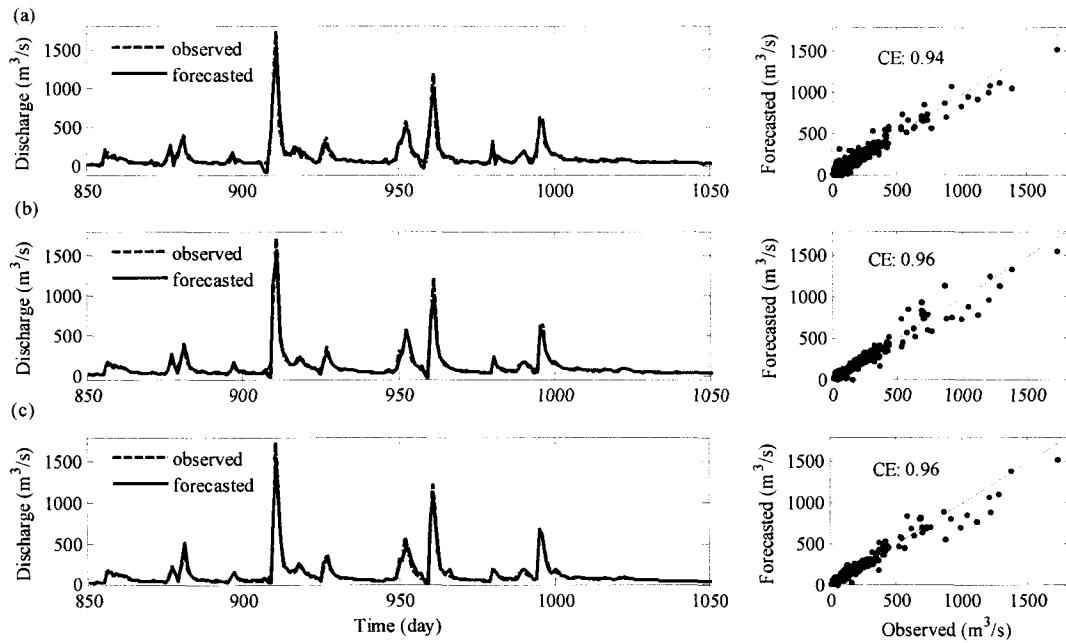


Figure 8.10 Hydrographs (representative details) and scatter plots of one-step-ahead prediction for Wuxi: (a) LR-SSA, (b) ANN-SSA, and (c) MANN-SSA

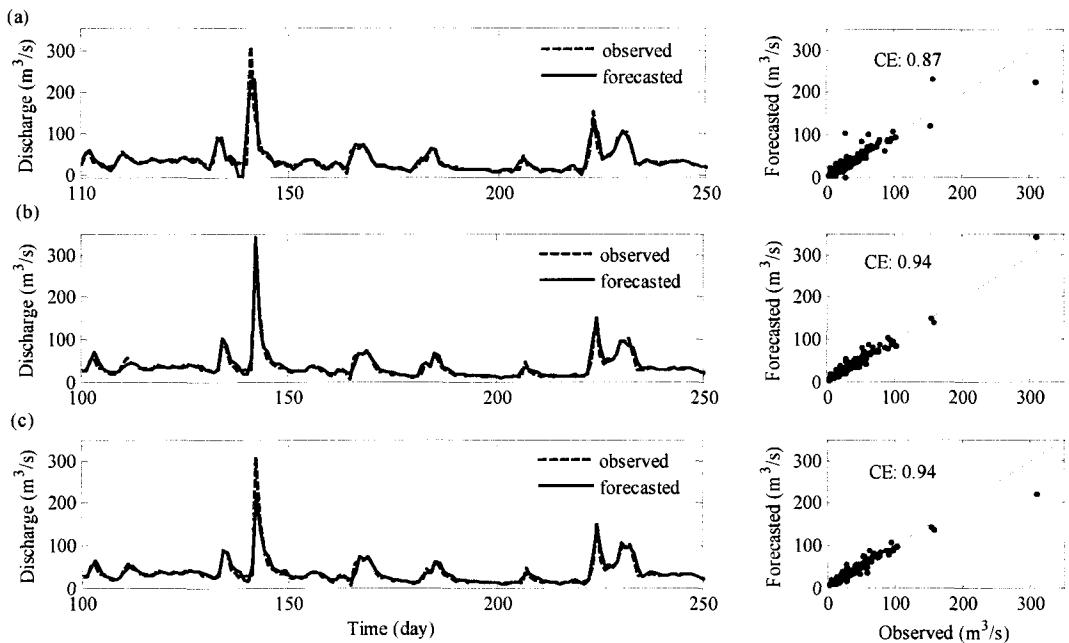


Figure 8.11 Hydrographs (representative details) and scatter plots of one-step-ahead prediction for Chongyang: (a) LR-SSA, (b) ANN-SSA, and (c) MANN-SSA

Table 8.8 presents comparison of two types of model inputs feeding ANN-SSA and MANN-SSA. ANN-SSA (or MANN-SSA) fed by rainfall and flow performed better

than the corresponding model fed by only flow at all prediction leads. It is observed that the advantage of models with rainfall and flow inputs over those with flow input only becomes more obvious with increasing prediction leads, which indicates that SSA improves the dependence relation between rainfall and flow more significantly than that between flows itself. The model output may therefore depend more on rainfall inputs instead of flow itself when the prediction lead is larger than one day.

Table 8.8 Performances of ANN-SSA and MANN-SSA using two types of input variables

| Watershed | Input variables | Model | RMSE | | | CE | | | PI | | | | | |
|----------------------|-----------------|-------|----------|-------|-------|-------|------|------|------|------|------|------|--|--|
| | | | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | | | |
| Wuxi | | | | | | | | | | | | | | |
| <i>Rainfall+Flow</i> | | | | | | | | | | | | | | |
| | | | ANN-SSA | 25.40 | 27.10 | 33.96 | 0.96 | 0.95 | 0.92 | 0.92 | 0.96 | 0.94 | | |
| | | | MANN-SSA | 25.08 | 26.87 | 34.05 | 0.96 | 0.95 | 0.92 | 0.92 | 0.96 | 0.94 | | |
| <i>Flow</i> | | | ANN-SSA | 31.02 | 50.64 | 61.80 | 0.94 | 0.83 | 0.74 | 0.88 | 0.84 | 0.80 | | |
| | | | MANN-SSA | 26.20 | 41.02 | 48.69 | 0.95 | 0.89 | 0.84 | 0.92 | 0.90 | 0.88 | | |
| Chongyang | | | | | | | | | | | | | | |
| <i>Rainfall+Flow</i> | | | | | | | | | | | | | | |
| | | | ANN-SSA | 6.22 | 7.08 | 11.12 | 0.94 | 0.93 | 0.82 | 0.91 | 0.93 | 0.86 | | |
| | | | MANN-SSA | 6.42 | 8.13 | 13.14 | 0.94 | 0.90 | 0.74 | 0.91 | 0.91 | 0.80 | | |
| <i>Flow</i> | | | ANN-SSA | 7.93 | 11.15 | 15.72 | 0.91 | 0.81 | 0.63 | 0.86 | 0.83 | 0.72 | | |
| | | | MANN-SSA | 7.32 | 10.19 | 15.71 | 0.92 | 0.84 | 0.63 | 0.88 | 0.86 | 0.72 | | |

Figure 8.12 illustrates one-step-ahead forecast hydrographs for Wuxi and Chongyang using ANN-SSA in two types of inputs. ANN-SSA with rainfall and flow inputs better captures the peak flows, and reproduces the actual hydrograph more smoothly whereas the hydrograph from ANN-SSA with flow input only is serrated at some locations. It is found that there is no time shift between the forecasted hydrograph and the actual one. Figure 8.13 demonstrates the results of lag effect analysis at all three prediction horizons. SSA eradicates the prediction lag effect in the ANN model regardless of model input types. However, it can be observed that the CCF curve in ANN-SSA with rainfall and flow inputs is more symmetrical than that in ANN-SSA with only flow input, which reveals that predictions in the former is in better agreement with the observations in time.

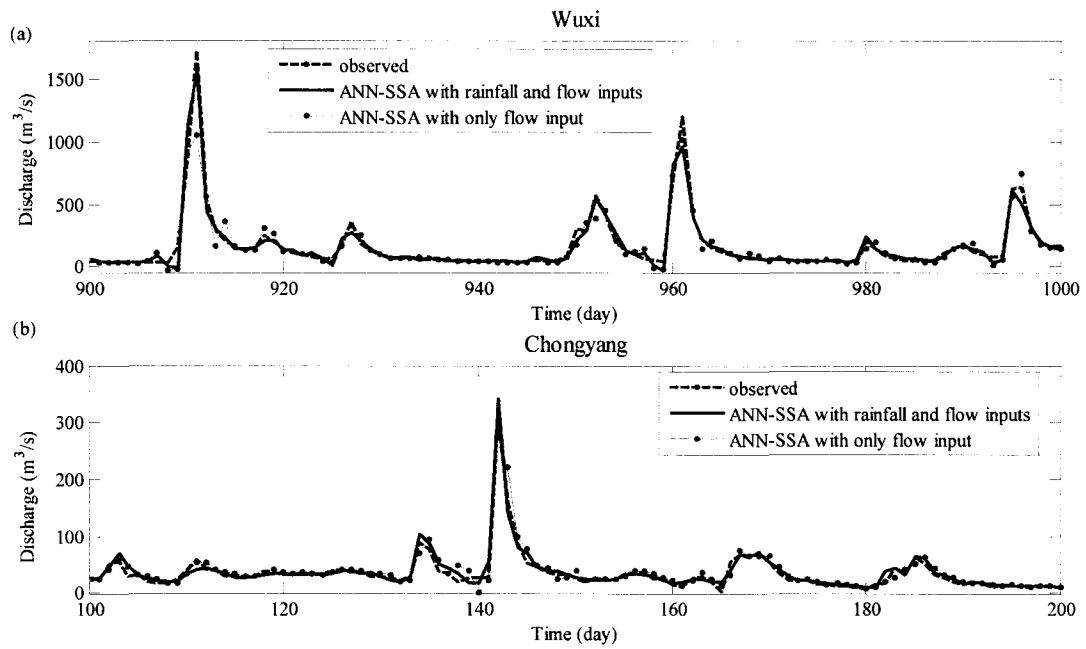


Figure 8.12 Hydrographs for one-step-ahead prediction using ANN-SSA with two types of inputs: (a) Wuxi, and (b) Chongyang.

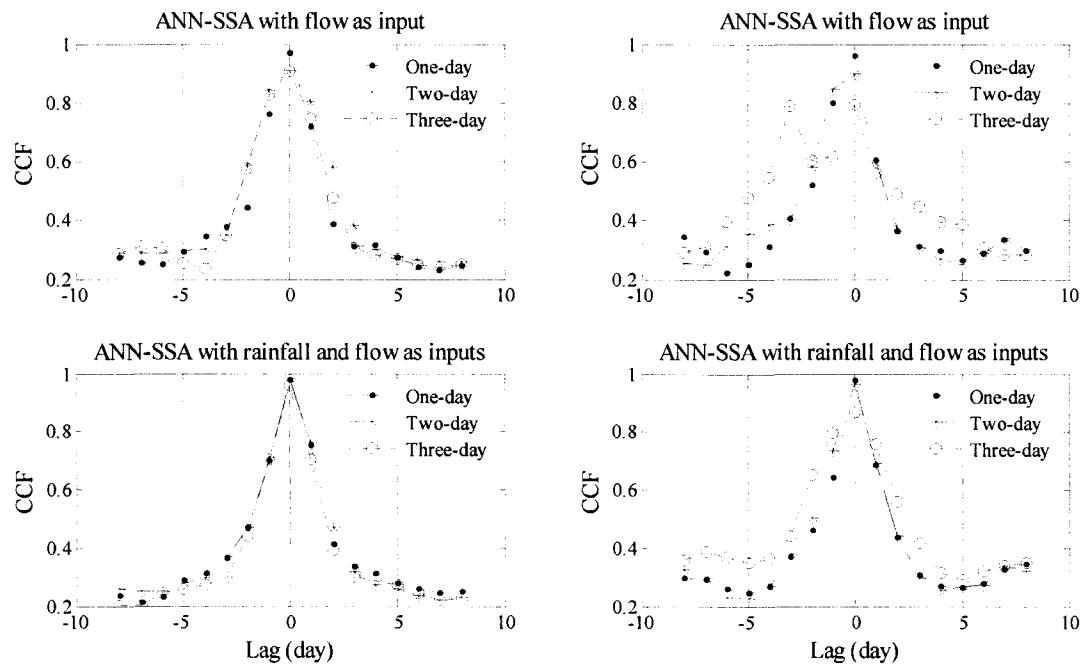


Figure 8.13 Lag analysis of observation and forecasts of ANN-SSA with two types of inputs: (a) and (c) for Wuxi, and (b) and (d) for Chongyang.

8.4.3 Discussions

The following discussions focus on two aspects: investigating the difference between two types of model inputs for streamflow forecasting, and investigating the effect of SSA on the R-R ANN model inputs.

a) Analysis of model inputs

As shown in Table 8.6, ANN with rainfall and flow inputs performs better than that with flow input only at all prediction leads, but the improvement of model performance decreases abruptly at a two-step lead. A direct explanation for that phenomenon is that the impact of rainfall on runoff weakens suddenly at two-step-ahead prediction, which can be examined by AMI and CCF between model inputs and output.

Figure 8.14 presents AMI between each input and output of ANN in two model input scenarios for the Wuxi study case. The number of model inputs in the abscissa axis is associated with that in Table 8.3 in the scenario of rainfall and flow as inputs. The former 5 inputs stand for 5 past flows and the latter 5 inputs denote 5 past rainfall observations. In contrast, all 10 model inputs (actually 5, referring to Chapter 7, here for convenience of plotting in the same figure) in the flow input scenario are the past 10 flow observations. First of all, it is clearly shown from all three sub-plots that AMI associated with each model input decreases significantly with an increase in the forecast lead, which may indicate decrease of the overall dependence relation between model inputs and output. Therefore, it provides a potential explanation for the trend in Table 8.6 that the accuracy of the forecast decreases with the increase of prediction horizons. Secondly, the nearest rainfall observation (the sixth model input in each plot) to the prediction horizon has the maximum AMI, so inclusion of such input improves the prediction. Some of the other rainfall inputs also have reasonably larger AMI compared to that of flow inputs, and they also contribute to the improvement of model performance.

Figure 8.15 shows AMI of each input and output of ANN with two types of inputs for the Danjiangkou study case. Regarding ANN in rainfall and flow inputs, the first 4 model inputs in the abscissa axis are from the past flows and the latter 5 inputs represent the 5 last rainfall observations. As far as ANN with flow input only is concerned, the first 4 model inputs in the abscissa axis are the actual inputs (referring to Chapter 7). It can be observed that, AMI of each model input and output between two-step-ahead and three-step-ahead forecasts is similar and very small regardless of the input scenario. Moreover, the holistic AMI from rainfall inputs does not dominate over the overall AMI from flow inputs. Therefore, inclusion of such rainfall inputs may only make the training process computation intensive without any tangible improvement in forecast accuracy. As a consequence, the model performance of ANN with two types of inputs is similarly poor for both two- and three-step-ahead forecasts (depicted in Table 8.6). On the contrary, for one-step-ahead forecast, the nearest two rainfall inputs have large AMIs which are only smaller than the AMI of the immediate past flow input. As expected, their inclusion in model inputs improves the overall mapping between inputs and output of ANN, making one-step-ahead prediction with good accuracy.

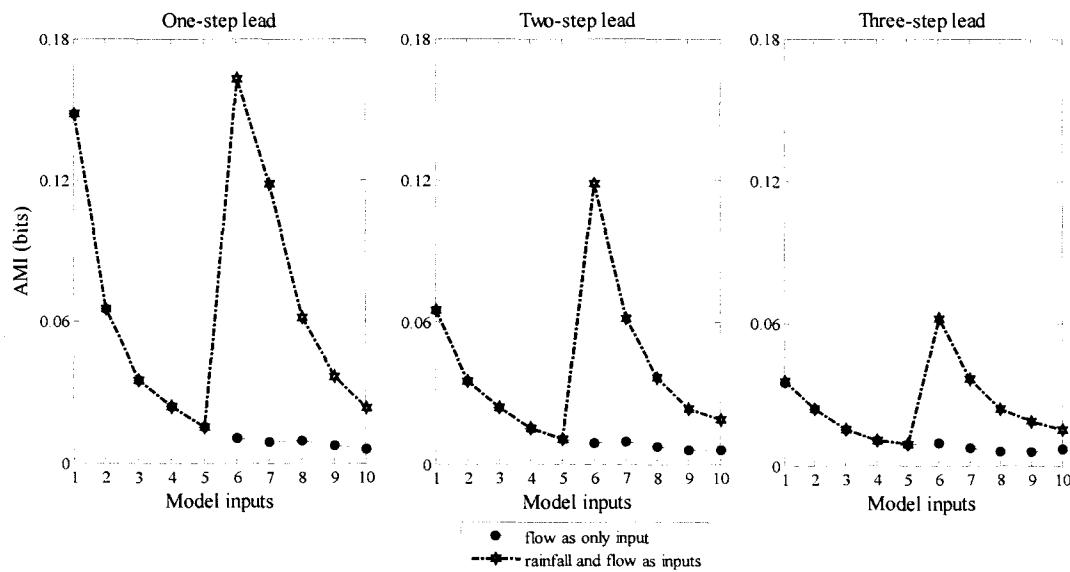


Figure 8.14 AMIs between model inputs and output for ANN with two types of inputs using the Wuxi data

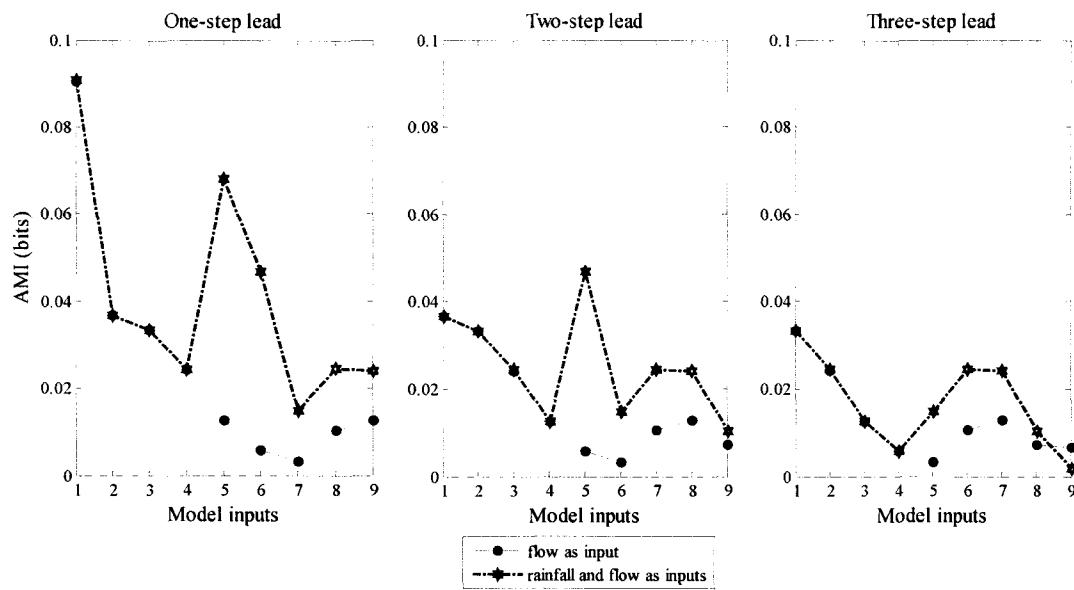


Figure 8.15 AMIs between model inputs and output for ANN with two types of inputs using the Chongyang data

The static multi-step prediction method is adopted in this study. The poor prediction at two- or three-step-ahead horizon using ANN with rainfall and flow as inputs may be improved by adopting a dynamic ANN model instead of the current static ANN model. In the dynamic ANN model, the forecasted flow and rainfall in the last step are used as the nearest flow and rainfall inputs in the present prediction step, and then a multi-step prediction becomes a repeated one-step prediction. However, de Vos and Rientjes (2005) mentioned that for both the daily and hourly data the two multi-step prediction methods performed nearly similar up to a lead time of respectively 4 days and 12 hours. Similarly, the results from Yu et al. (2006) for hourly data also showed that two methods could yield similar forecasts.

b) Investigation of the SSA effect on model inputs

Herein, the effect of SSA on inputs of an ANN R-R model is investigated by AMI between each input and output of model. Results of forecast from the ANN R-R model with the normal mode (shown in Table 8.5 or Table 8.6) indicate that the flows at one-step lead are predicted appropriately whereas poor forecasts are obtained at two- or three-step lead. Correspondingly, it can be observed from Figure 8.16 that AMI associated with each model input for one-step forecast is far larger than the

counterpart for two- or three-step forecasts. Figure 8.17 shows that SSA improves AMI of each input at all three prediction horizons. The AMI curve of filtered inputs between one- and two-step forecasts is very similar, which may indicate similar model performance (shown in Table 8.7 or Table 8.8 where the model performance at the two prediction leads is also quite similar). Therefore, the AMI analysis proves to be able to reveal the suitability of a forecasting model to some extent. Figure 8.17 also reveals that AMI at one-step forecast is far larger than that at two- and three-step leads. So the prediction accuracy at the former is markedly superior to that in the latter (shown in Table 8.5 or Table 8.6). In the SSA mode, AMI of each input improves considerably at all prediction horizons, which renders the ANN-SSA R-R model possible to obtain good predictions (shown in Table 8.7 or Table 8.8) in comparison to that in the normal mode.

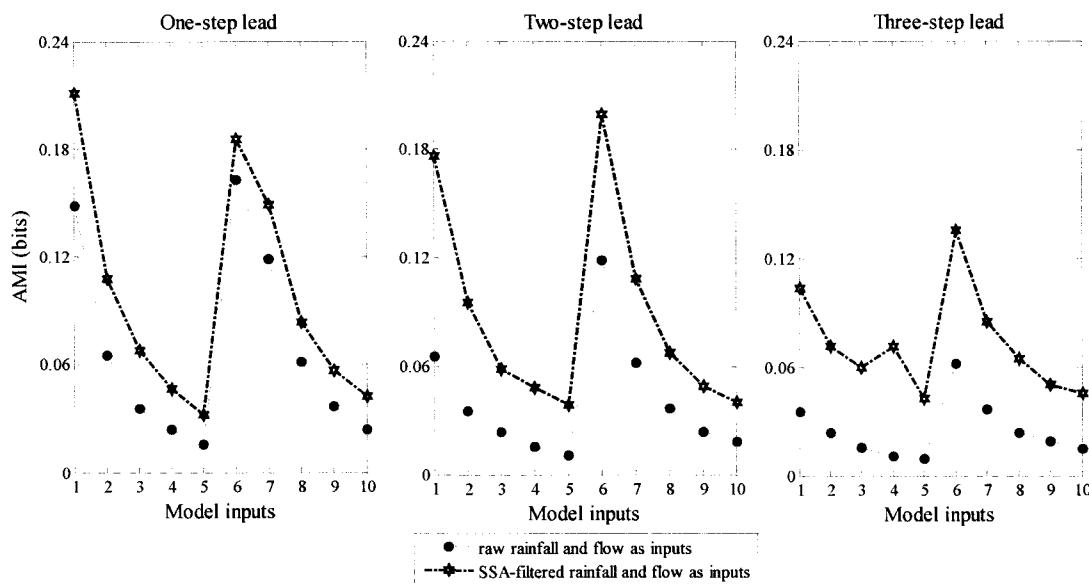


Figure 8.16 AMIs between model inputs and output for ANN and ANN-SSA in the context of R-R forecasting using the Wuxi data

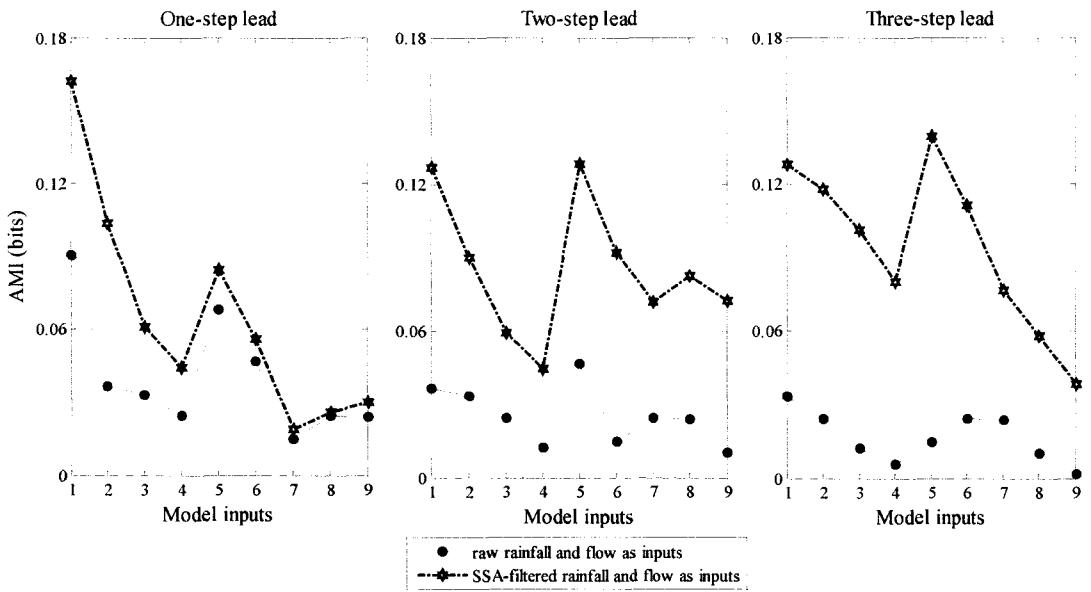


Figure 8.17 AMIs between model inputs and output for ANN and ANN-SSA in the context of R-R forecasting using the Chongyang data

8.5 Summary

This chapter has investigated daily streamflow predictions in the context of rainfall-runoff transformation in comparison to the results from Chapter 7. Three models including LR, ANN and MANN are conducted in both the normal mode and SSA mode using two case studies. The key points can be summarized as follows:

Rainfall and flow are identified as appropriate input variables, and then the model inputs are finally selected by LCA after comparison with the other four methods.

The model performance appears to be sensitive to studied cases in the normal mode. For Wuxi, the MANN R-R model (namely, rainfall and flow as inputs) outperforms the ANN R-R model and the ANN R-R model performs better than the LR R-R model at all three prediction horizons. For Chongyang, the ANN R-R model performs the best at one-step lead. However, they are similar at the other two prediction horizons. In the SSA mode, the performance of each model is improved significantly. Both ANN-SSA and MANN-SSA have similar performance and

achieve better results than LR-SSA. In view of ANN's relative ease in establishing rainfall/runoff mapping than MANN, proposed R-R forecasting model is therefore ANN coupled with SSA.

The ANN R-R model is also compared with the ANN model with only flow input (depicted in Chapter 7) in the normal mode and SSA mode. Irrespective of modes, the ANN R-R model outperforms the ANN model with only flow input. The degree of superiority tends to mitigate with the increase of forecast leads in the normal mode. However, situation becomes reverse in the SSA mode where the advantage of the ANN R-R model seems to be more remarkable with the increase of prediction leads.

9 Uncertainty Forecasting

Previous predictions on rainfall and streamflow are deterministic, i.e. in the form of a point forecast which does not take into account of various sources of uncertainties including model uncertainty, input uncertainty, and parameter uncertainty. Actually, incorporating prediction uncertainties into deterministic forecasts helps enhance the reliability and credibility of the model outputs. Corresponding to point forecasts, uncertainty forecasts are termed interval predictions in the literature. In order to assess the forecast quality of the ANN-SSA R-R model in Chapter 8, interval predictions will be herein conducted by the method of UNEEC (Shrestha and Solomatine, 2006; Solomatine and Shrestha, 2009).

9.1 Introduction

Uncertainty has always been inherent in water resources engineering and management. For example, in river flood defenses it was treated implicitly through conservative design rules, or explicitly by probabilistic characterization of meteorological events leading to extreme floods (Solomatine and Shrestha, 2009). In hydrological modeling, model errors are inevitable owing to the inherent uncertainties in the process. These uncertainties are strongly related to our understanding and measurement capabilities on the real-world system under study. Three important uncertainty sources have been recognized (Gupta et al., 2005): (1) uncertainties in the training data (or calibration data) (e.g., precipitation, evaporation, streamflow); (2) uncertainties in model parameters; and (3) uncertainties due to imperfect model structure.

A number of methods have been proposed in the literature to estimate uncertainty of the model output. According to Shrestha and Solomatine (2006), these methods are summarized into four categories: (1) probabilistic forecasting method (Krzysztofowicz, 2000); (2) method based on the analysis of model errors

(Chryssolouris et al., 1996; Heskes, 1997; Montanari and Brath, 2004); (3) simulation and re-sampling based methods (Beven and Binley, 1992; Kuczera and Parent, 1998); and (4) method based on fuzzy theory (Maskey et al., 2004). However, each of these methods has noteworthy drawbacks. For example, the first and third methods analyze the uncertainty of the uncertain input variables or data by propagating it through the deterministic model to the outputs, and hence require their priori distributions (generally taking assumptions). The second method requires certain assumptions regarding the residuals and data (e.g., normally and homoscedasticity). Evidently, the relevancy and accuracy of such methods rely on the validity of these assumptions. The last method entails knowledge of the membership function of the quantity subject to the uncertainty which could be very subjective. Furthermore, the majority of these methods deal only with a single source of uncertainty (Solomatine and Shrestha, 2009). For instance, the Monte Carlo-based approach from the third category tends to analyze the uncertainty source independently. The method based on the analysis of model errors typically computes the uncertainty of the “optimal model” (i.e. the model with uniquely optimal model parameters), and not of the “class of models” (i.e., the same structure but equifinal mode parameters).

However, it is more important to know the total model uncertainty accounting for all sources of uncertainty than the uncertainty resulting from individual source in the decision-making process (Solomatine and Shrestha, 2009). Recently, Shrestha and Solomatine (2006) developed a novel method to estimate the uncertainty of the “optimal model” that takes into consideration the joint contribution of all sources of errors. This method is referred to as an “uncertainty estimation based on local errors and clustering” (UNEEC). It may fall into the second category mentioned above since this method also assumes the model error to be an indication of model uncertainty. UNEEC utilizes the FCM clustering and machine learning techniques (ANN in the current study) to estimate the uncertainty of the model (the ANN-SSA R-R model herein) output by analyzing forecast residuals (errors). As pointed out by Solomatine and Shrestha (2009), the UNEEC has several advantages over commonly-used methods mentioned above. It is not imperative to make any assumption about residuals since the probability density function (pdf) of the model

error is estimated via empirical distribution. Moreover, the method is computationally efficient, and therefore can be easily applied to computationally demanding process models.

The purpose of the present study is that employing the UNEEC method conducts the uncertainty estimates of the ANN-SSA R-R model. Two daily streamflow study cases, Wuxi and Chongyan, are explored here. The optimal model structure of ANN-SSA is carried directly from the results in Chapter 8. For an evaluation purpose, the UNEEC method is compared to that produced by the bootstrap method on testing data (re-sampling techniques) which is widely used for ANN (e.g., Chryssolouris et al., 1996; Tibshirani, 1996; Heskes, 1997). This chapter is organized as follows. Section 9.2 presents methodology of the UNEEC method. In the section 9.3, main results are shown with necessary discussions. Section 9.4 summarizes main points in this chapter.

9.2 Methodology

9.2.1 Case studies

Two river basins, Daning and Lushui, are considered as case studies. They are respectively referred to as “Wuxi” and “Chongyang” named after the hydrology station at the outlet of the studied drainage area. More detailed descriptions on them can be found in Chapter 8. The present study attempts to extend point predictions of the ANN-SSA R-R model in chapter 8 to interval predictions. Hence, the point forecasting results are carried directly from Chapter 8.

9.2.2 Prediction interval

An interval prediction is usually expressed in the form of the upper and lower limits between which a point forecast is expected to lie with a specified probability. This limit is termed prediction limit or bound (PL), while the interval is termed the prediction interval (PI) (depicted in Figure 9.1). The specified probability is called

confidence level. It is worth noting that the PI is not equivalent to the confidence interval (CI). The CI is related to the accuracy of our estimate of the true regression whereas the PI deals with the accuracy of our estimate with respect to the observed target value. Clearly, the PI is wider than the CI (Heskes, 1997). Therefore, PI is of more practical use than CI because prediction interval is concerned with the accuracy with which we can predict the observed target value itself, but not just the accuracy of our estimate of the true regression (Shrestha and Solomatine, 2006).

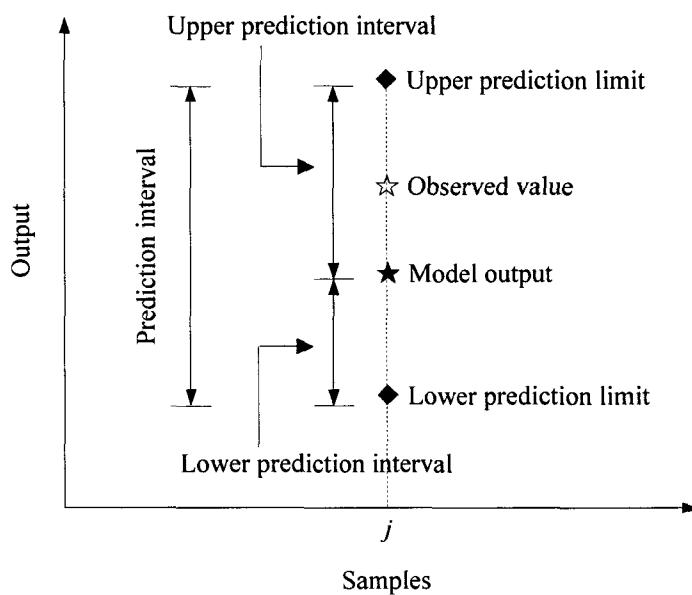


Figure 9.1 Terminology used in this chapter (adopted from Shrestha and Solomatine, 2006)

9.2.3 UNEEC

Figure 9.2 illustrates the generalized framework of the UNEEC method consisting of three sequential steps (bounded in dashed boxes): point prediction, estimates of upper and lower PIs, and interval prediction. The interval prediction is the summation of the former two. Thus, estimate of PI is the central task in this section because point prediction has been conducted in Chapter 8. The estimate of PI is completely based on operations of model errors. A brief explanation of model errors is first of all presented as follows (Solomatine and Shrestha, 2009).

A deterministic model M of a real-world system predicting a system output variable y^* given input vector x ($x \in \mathbf{X}$) is considered. Let y be the measurement

(observation) of an unknown true value y^* , made with error e_y . Various types of errors propagate through the model M while predicting the observed output y and have the following form:

$$y = y^* + e_y = M(\mathbf{x}, \theta) + e_s + e_\theta + e_x + e_y \quad (9.1)$$

where θ is a vector of the model parameters values, e_s , e_θ , and e_x are the errors associated with the model structure M , parameter θ and input vector \mathbf{x} , respectively. The contribution of each error component to the model error is typically not known and, as pointed out by Gupta et al. (2005), disaggregation of errors into these components is often difficult, particularly in hydrology where models are nonlinear and these error sources may interact to produce the measured deviation. Thus, the different components that contribute to the total model error are generally treated as a single lumped variable and Eq. (9.1) can be reformulated as

$$y = \hat{y} + e \quad (9.2)$$

where \hat{y} is the model output and e is the total residual error. Thus, the UNEEC method estimates the lower and upper PIs associated with the given model structure M and parameter set θ by analyzing historical model residuals e which is a combined effect of all sources of error.

As shown in Figure 9.2, the estimate of PI consists of three main parts: (1) clustering; (2) computing PIs for each training sample; and (3) constructing model for the PI estimate.

Clustering of data is an important step of the UNEEC method. The data are comprised of part or all of model inputs corresponding to model errors. The most relevant inputs are determined by the AMI analysis between model inputs \mathbf{X} and model errors e . The selected inputs are denoted as \mathbf{X}_c where the subscript stands for clustering. As shown in Chapter 8, the model inputs \mathbf{X} consist of various lags of rainfall and flow variables in the context of the R-R forecasting. By applying the FCM clustering method to \mathbf{X}_c , these errors e can be partitioned into c clusters. The clustering is based on a strong assumption: the input data which belong to the same cluster will have similar characteristics and corresponding to similar real-life

situations; Furthermore, the distributions of the model errors within different clusters have different characteristics. The assumption has been partly verified in hydrology community where modular models are capable of making more robust forecasts than a global model.

Having identified the clusters, a succeeding task is to compute PIs for each cluster first, and then for each sample. The PIs for each cluster are computed from empirical distributions of the corresponding model errors on the training data (or calibration data). For instance, in order to construct $100(1-\alpha)\%$ PI, the $(\alpha/2) \times 100$ and $(1-\alpha/2) \times 100$ percentile values are taken from empirical distribution of residuals for lower and upper prediction interval, respectively. Typical value for α is 0.05, which corresponds to 95% prediction interval. In the context of fuzzy clustering, each sample belongs to more than one cluster and is associated with several membership grades, the computation of the above percentiles should take this into account. However, this computation is very straightforward if the k-means clustering (a crisp clustering method) is used for the split of the input space. To calculate PI, the samples should first be sorted with respect to the corresponding errors in ascending order. The following expression gives the lower prediction interval for cluster i (PIC^L)

$$PIC_i^L = e_j, \quad j : \sum_{k=1}^j \mu_{i,k} < \alpha/2 \sum_{j=1}^N \mu_{i,j} \quad (9.3)$$

where j is its maximum value that satisfies the above inequality (each side of the inequality reflects cumulative probability density taking membership degree into consideration), e_j is the error associated with the sorted sample j , $\mu_{i,j}$ is the membership degree of j th sample to cluster i , N is the number of samples in the input space. Similar expression can be obtained for the upper PI (PIC^U) when substituting $1-\alpha/2$ for $\alpha/2$ in Eq. (9.2).

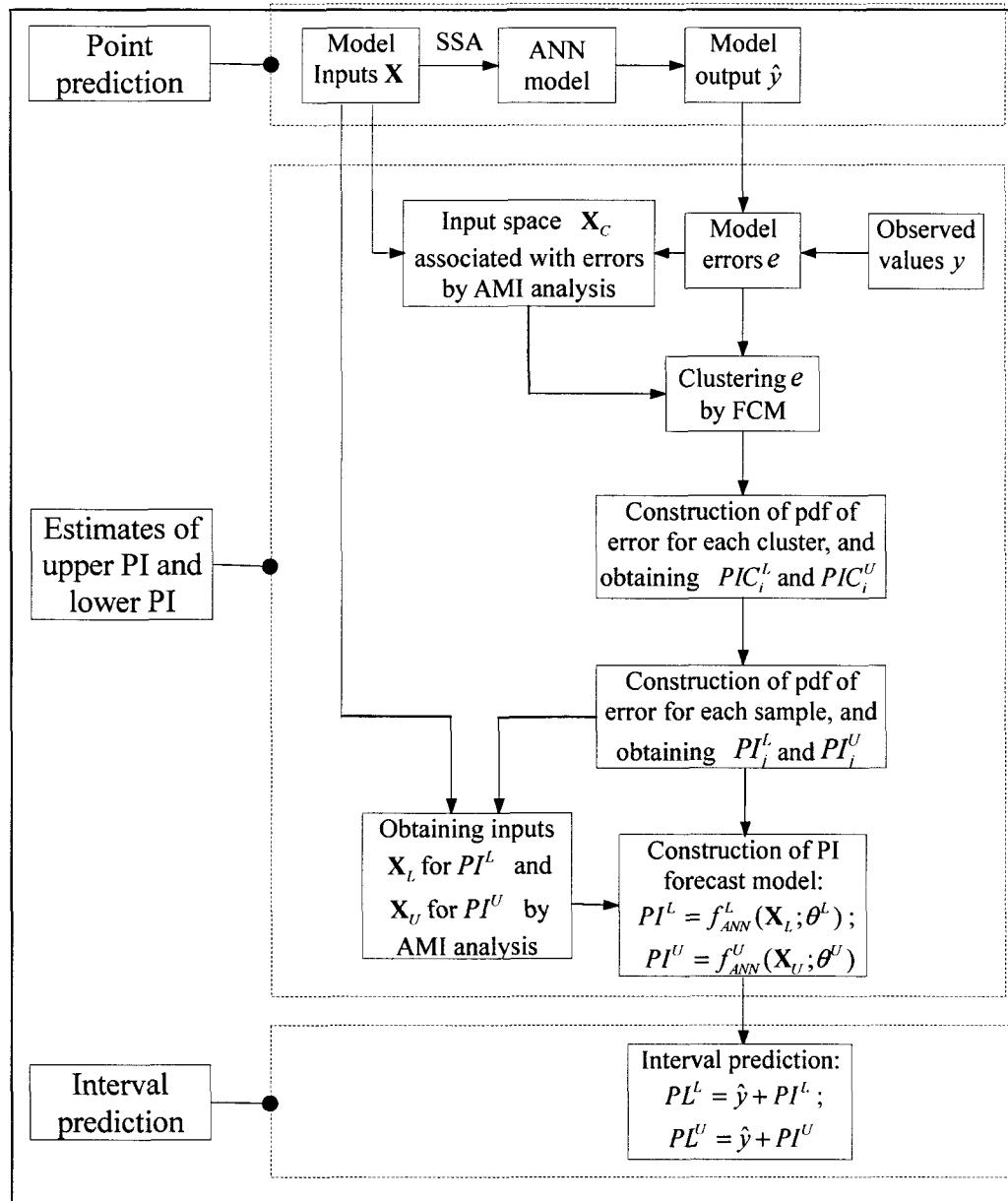


Figure 9.2 The generalized framework of the UNEEC method

Once the PI is computed for each cluster, the PI for each sample in input space can be computed by the clustering technique. For example, if crisp clustering is employed, then the PI for each sample in the particular cluster is the same as that of the cluster. In contrast, in the case of fuzzy clustering, the PI is computed using the weighted mean of PI of each cluster as

$$PI_j^L = \sum_{i=1}^c \mu_{i,j} PIC_i^L, \text{ and } PI_j^U = \sum_{i=1}^c \mu_{i,j} PIC_i^U \quad (9.4)$$

where PI_j^L and PI_j^U are the lower and upper prediction interval for j th sample, respectively.

Once the lower and upper prediction interval corresponding to each sample is obtained, two independent ANN model can be established to estimate the underlying functional relationships between the input vector and prediction intervals as

$$PI^L = f_{ANN}^L(\mathbf{x}_L; \theta^L), \text{ and } PI^U = f_{ANN}^U(\mathbf{x}_U; \theta^U) \quad (9.5)$$

where θ^L and θ^U are the parameter vectors of the ANN model for the PI^L and PI^U , respectively, \mathbf{x}_L and \mathbf{x}_U are the model inputs for the PI^L and PI^U , and they can be determined by the AMI analysis between the initial model input \mathbf{x} and the PI (respectively using PI^L and PI^U).

Once $f_{ANN}^L(\cdot)$ and $f_{ANN}^U(\cdot)$ are trained on the training data, it can be employed to estimate the prediction intervals for the new data input. Thus, the interval prediction can be obtained by simply adding the model output to the prediction intervals as

$$PL^L = \hat{y} + PI^L, \text{ and } PL^U = \hat{y} + PI^U \quad (9.6)$$

where PL^L and PL^U are the lower and upper prediction limits.

9.2.4 Performance evaluation of UNEEC

Three measures are employed for the performance evaluation of UNEEC. The first one is involved in evaluating the prediction interval coverage probability (PICP). The PICP is the probability that the observed value of an input pattern lies within the prediction limits and is estimated by the corresponding frequency as follows

$$PICP = \frac{1}{V} \text{count } j, \text{ where } j : PL_j^L \leq y_j \leq PL_j^U \quad (9.7)$$

where V is number of data in the test set, y_j represents the j th observed value. If the clustering technique and the UNEEC model are optimal, then the PICP value will be consistently close to the $(1 - \alpha/2)\%$.

The second measure is the mean prediction interval (MPI) calculated across all

points in the testing set. It measures the ability to enclose observed values inside the prediction bounds and can be estimated by

$$MPI = \frac{1}{V} \sum_j^V (PL_j^U - PL_j^L) \quad (9.8)$$

The last one is to examine relative MPI (i.e., the ratio of MPI to the average of observed values) and expressed as

$$RMPI = \frac{\frac{1}{V} \sum_j^V (PL_j^U - PL_j^L)}{\bar{y}} \quad (9.9)$$

where \bar{y} denotes average observed value. It reflects the quality of identified prediction bounds. In general, when PICP is similar, the smaller RMPIT is, the better overall quality prediction intervals have.

9.3 Results and Discussions

9.3.1 Analysis of model errors

The model errors of ANN-SSA and their norm-plots from Wuxi and Chongyang are shown in Figure 9.3. It can be observed that the errors seem to be highly correlated with the observed flows because the overall trend between flows and mode errors is quite similar, viz. the model errors increase with the increase of flows, particularly for Wuxi. This also indicates the presence of heteroscedasticity in the residuals. The normal probability plots of the residuals (or errors) show that the residuals are not normally distributed because their probability coordinates are far from the straight line which represents a standard normal distribution. Therefore, the identified ANN-SSA may need to be further optimized for reducing model residuals from model structure or parameters as much as possible. Certainly, it is often difficult to find a model having a normal distribution of errors, which renders traditional errors-based uncertainty methods not feasible.

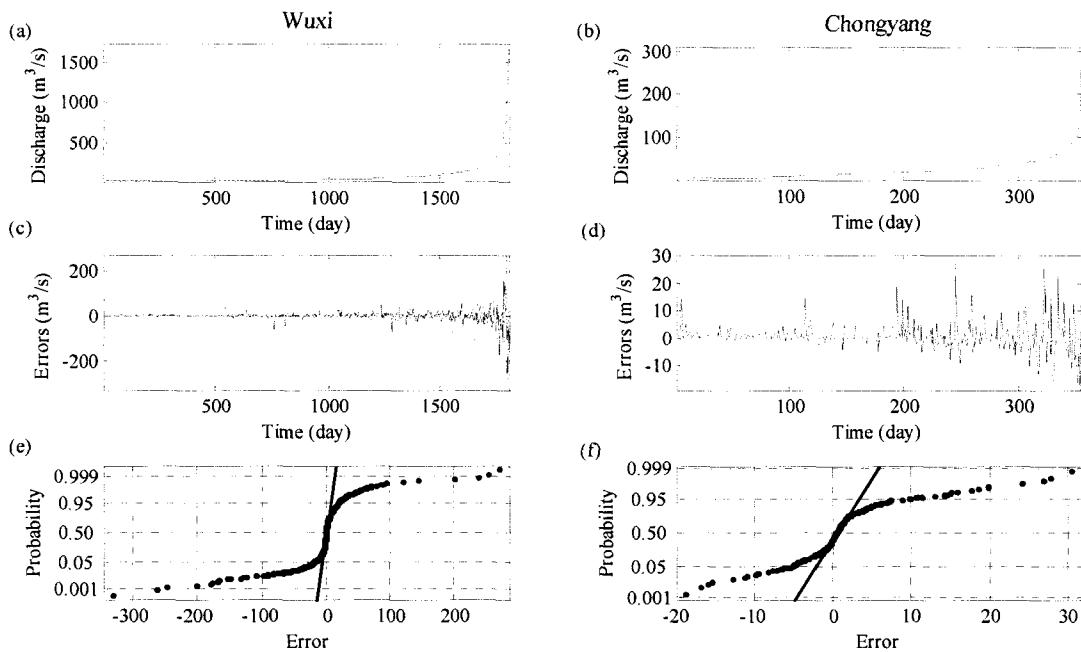


Figure 9.3 Original flow, model residual errors, and norm-plots of errors for Wuxi (the left column) and Chongyang (the right column)

9.3.2 Clustering

In the process of clustering of errors, the input variables and the optimal number of clusters are required. In this study, the input variables are chosen by the AMI analysis between raw model inputs and model residuals. The results of AMI for Wuxi and Chongyang are presented in Figure 9.4. Herein, the top four variables in a descending order of AMIs are considered forming \mathbf{X}_c . Referring to Table 8.3, four inputs in \mathbf{X}_c are $Q_{t-4}, R_{t-4}, Q_{t-3}$, and Q_{t-2} for Wuxi, and Q_{t-1}, Q_{t-3}, R_t , and R_{t-2} for Chongyang, respectively. It is worthwhile to note that the results in Figure 9.4 may be changeable due to the instability of the output of ANN.

The number of clusters is determined by trial and error with varied c from 2 to 6. Table 9.1 depicts the sensitivity of uncertainty evaluation indices to the number of clusters c . In the case of Wuxi, the PICP appears to be insensitive to c . With similar PICP, the smaller MPI or PRMIT is, the better the performance of UNEEC is. Therefore, the optimal number of clusters is set at the value of 5. For Chongyang, the optimal number of clusters is set at the value of 4 on the basis of a compromise

amongst three performance indices. These results are similar to those in Solomatine and Shrestha (2009) where the UNEEC method was also used in a rainfall-runoff prediction.

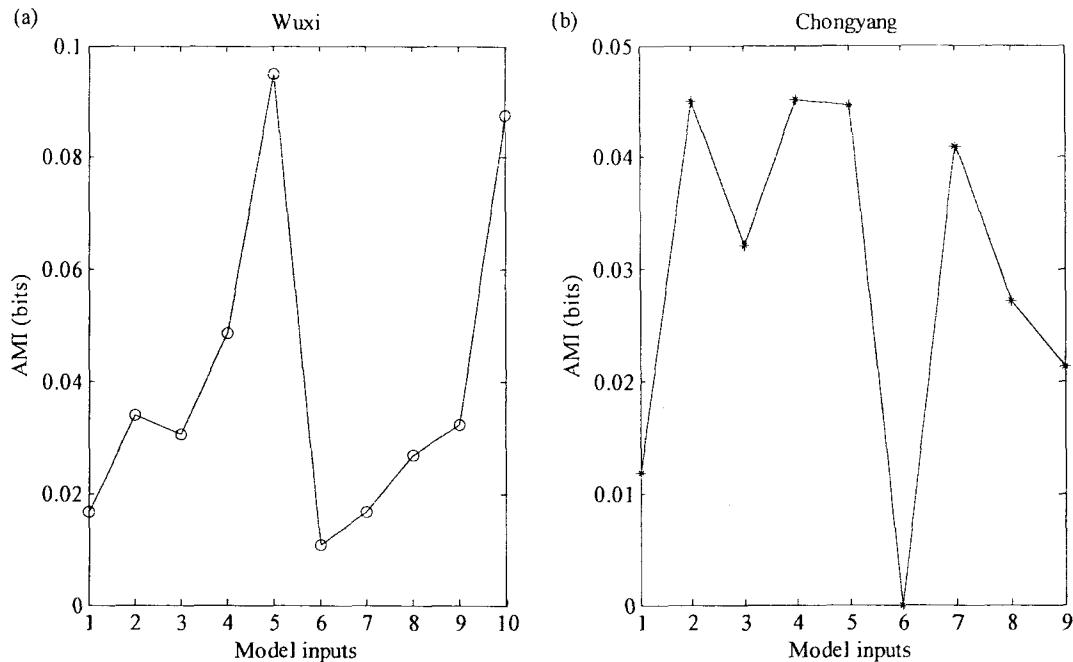


Figure 9.4 AMI between raw model inputs and model errors: (a) Wuxi, and (b) Chongyang.

Table 9.1 Sensitivity of uncertainty evaluation indices to number of clusters

| watershed | Number of clusters | PICP (%) | MPI (m^3/s) | RPMI |
|------------------|--------------------|-------------|-------------------------------|-------------|
| Wuxi | | | | |
| | 2 | 94.8 | 52.60 | 0.79 |
| | 3 | 94.7 | 53.44 | 0.81 |
| | 4 | 94.7 | 48.26 | 0.73 |
| | 5 | 94.7 | 49.23 | 0.74 |
| | 6 | 94.9 | 50.21 | 0.76 |
| Chongyang | | | | |
| | 2 | 93.8 | 24.26 | 0.97 |
| | 3 | 95.2 | 22.54 | 0.91 |
| | 4 | 95.5 | 23.49 | 0.94 |
| | 5 | 94.9 | 21.53 | 0.87 |
| | 6 | 94.9 | 21.96 | 0.88 |

9.3.3 Model identification for lower and upper PIs

ANN is used as the mapping function to capture the relation between potential causal factors and the PI. These factors are selected from the raw model inputs by AMI analysis. The present study took α as the value of 0.05, and hence quantiles for lower and upper PIs are respectively set at 2.5% and 97.5%. Each PI was computed according to Eqs. (9.3) and (9.4). Figure 9.5 illustrates the AMI results. It can be observed that the AMI curves from the lower PI and upper PI have a similar trend, which means the correlations between both of them and raw model inputs are similar. In the process of setting up the ANN model, only four inputs are selected which are associated with the top four largest AMIs. The identified inputs are presented in Table 9.2.

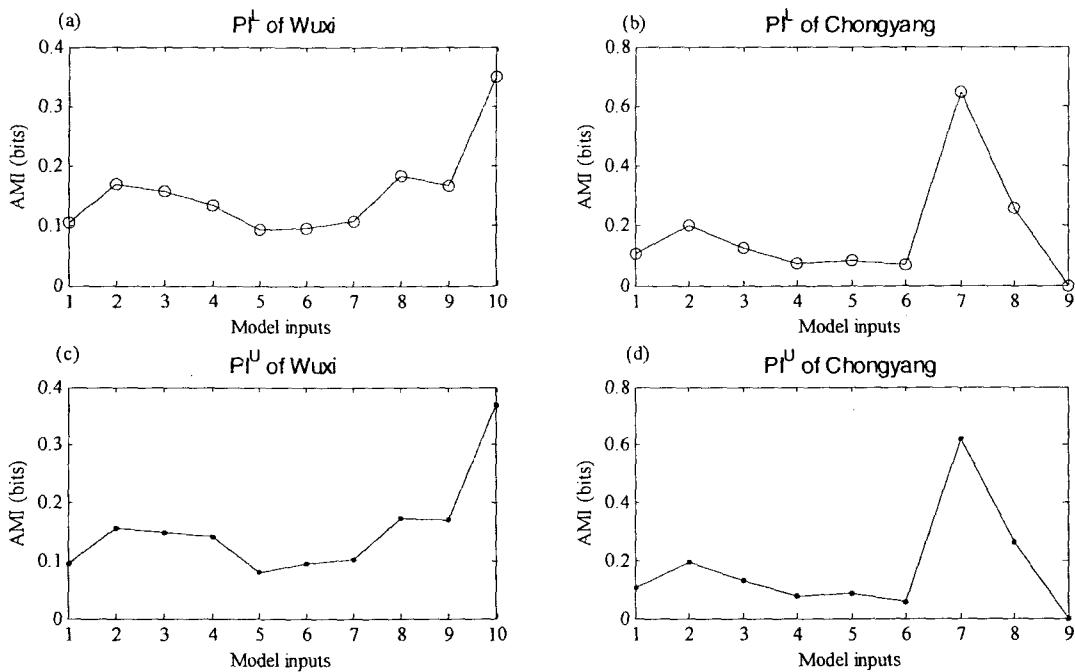


Figure 9.5 AMI between raw model inputs and PI: (a) and (c) for Wuxi, and (b) ad (d) for Chongyang

Having the input and output data pair, the identification of ANN consists in finding the optimal nodes of hidden layer when a three-layer perceptron is employed. The partition of the data pair into training, cross-validation, and testing subsets is in line with that in the point prediction model (described in Chapter 8). The optimal number

of nodes is found by systematically increasing the number of hidden neurons from 1 to 10 until the ANN performance on the cross-validation set no longer improves significantly. Relevant information of these models is shown in Table 9.2.

Table 9.2 Relevant information of ANN models for PIs

| Watershed | Type of PI | Model inputs | | | | Model structure |
|------------------|-----------------|--------------|-----------|-----------|-----------|-----------------|
| | | 1 | 2 | 3 | 4 | |
| Wuxi | | | | | | |
| | PI ^L | R_{t-4} | R_{t-3} | R_{t-2} | Q_{t-1} | 4-5-1 |
| | PI ^U | R_{t-4} | R_{t-3} | R_{t-2} | Q_{t-1} | 4-4-1 |
| Chongyang | | | | | | |
| | PI ^L | R_{t-2} | R_{t-3} | Q_{t-1} | Q_{t-2} | 4-7-1 |
| | PI ^U | R_{t-2} | R_{t-3} | Q_{t-1} | Q_{t-2} | 4-7-1 |

9.3.4 Analysis of the model uncertainty

Figure 9.6 and Figure 9.7 show 95% prediction limits and prediction intervals respectively for Wuxi and Chongyang when using the UNEEC method for uncertainty analysis at one-day lead. The plot (b) in Figure 9.6 or Figure 9.7 results from plot (a) subtracted by observations for the purpose of visual inspection. It can be clearly observed that the PLs almost enclose all model errors. The plot (c) in each Figure shows that the fluctuation process of the PI curve is highly consistent with that in the original flow series in the plot (a), namely, the high flows have large PIs and low flows have small PIs.

Table 9.3 presents performance measures of UNEEC for the testing data of Wuxi and Chongyang compared with the bootstrap method. The value of PICP from UNEEC is very close to the desired value of 95% for each study case. However, it can be noticed that only about 65% of the observed flow values fall inside the 95% prediction limits estimated by the bootstrap method. The bootstrap method follows the following procedures: the bootstrap of the testing data was first implemented, and then prediction on each bootstrapped testing data is conducted using the same model structure identified in the stage of training. Clearly, the bootstrap method only accounts for input uncertainty whereas the UNEEC method assumes the use of the optimal model and treats all other sources of uncertainty in an aggregated form.

Therefore, as shown in Table 9.3, the MPI from the bootstrap method is certainly narrower than that produced by the UNEEC. As far as the bootstrap method is concerned, more points are below the lower PL, which may mean the lower PI is much underestimated than the upper PI.

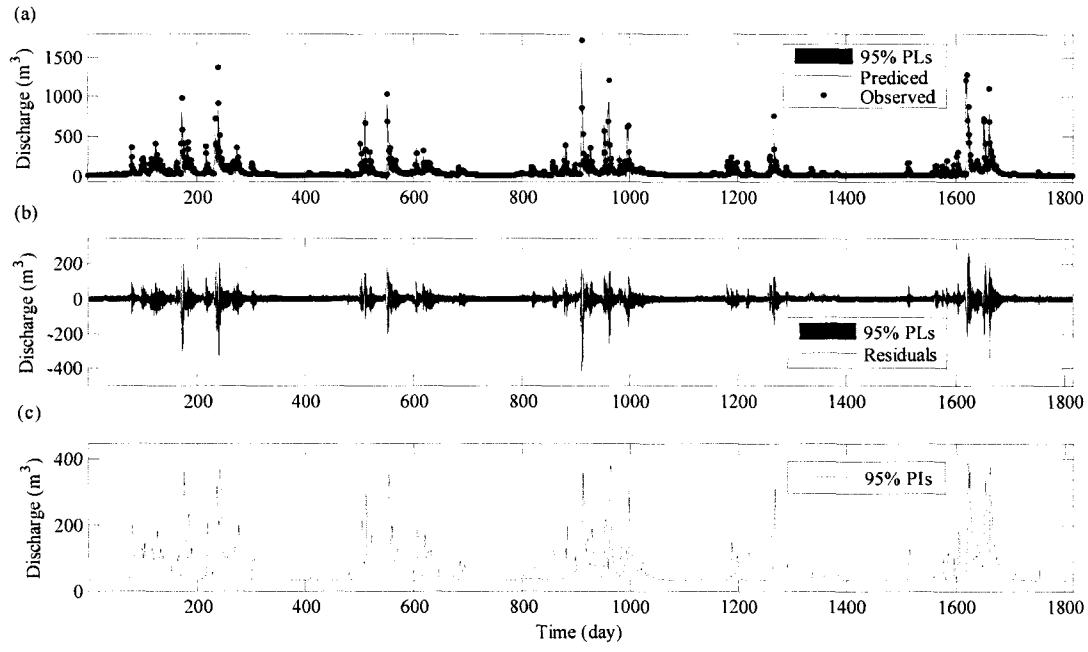


Figure 9.6 95% prediction limits (a and b) and 95% prediction intervals (c) for Wuxi test data using ANN model to predict one-day lead

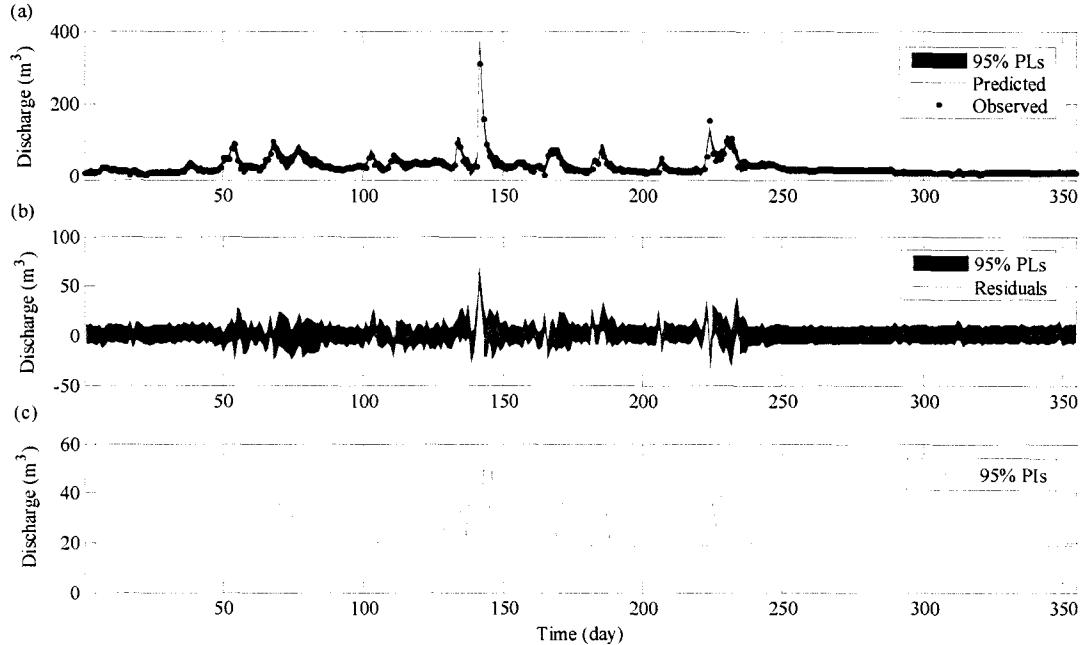


Figure 9.7 95% prediction limits (a and b) and 95% prediction intervals (c) for Chongyang test data using ANN model to predict one-day lead

A further analysis on the distribution of these points outside the PIs is presented in Table 9.4 where the raw flows is categorized into low and high flows using half of maximum flow as the dividing line. It can be observed that less high flows are outside the PIs in the bootstrap method than those in the UNEEC. The bootstrap method better estimates the uncertainty of high flows, but lacks the capability in uncertainty prediction for the low flows. On the contrary, the UNEEC method proves to be capable of making proper uncertainty estimates on low flows.

Table 9.3 Comparison of UNEEC and Bootstrap

| Watershed | Method | Performance measure | | | Points out of PIs | |
|------------------|------------|---------------------|-----------------|------|---------------------------|---------------------------|
| | | PICP (%) | MPI (m^3/s) | RPMI | Below PL ^L (%) | Above PL ^U (%) |
| Wuxi | | | | | | |
| | UNEEC | 94.7 | 48.26 | 0.73 | 1.98 | 3.32 |
| | Bootstrap* | 64.6 | 17.82 | 0.27 | 21.7 | 13.7 |
| Chongyang | | | | | | |
| | UNEEC | 94.9 | 21.53 | 0.87 | 3.80 | 1.30 |
| | Bootstrap* | 65.4 | 6.65 | 0.27 | 20.6 | 14.1 |

Note: * B for bootstrap is taken the value of 1000.

Table 9.4 Analysis of flow compositions

| Watershed | Flow types | Distribution of flows (%) | |
|------------------|-------------------------------|---------------------------|-----------|
| | | Low flow | High flow |
| Wuxi | | | |
| | Raw flow series | 99.39 | 0.61 |
| | Flow outside PIs by UNEEC | 4.75 | 0.55 |
| | Flow outside PIs by Bootstrap | 35.21 | 0.17 |
| Chongyang | | | |
| | Raw flow series | 99.44 | 0.56 |
| | Flow outside PIs by UNEEC | 5.04 | 0.28 |
| | Flow outside PIs by Bootstrap | 34.65 | 0.00 |

For convenience of visual inspection, representative details of uncertainty predictions from the UNEEC and bootstrap are presented in Figure 9.8 for Wuxi and Figure 9.9 for Chongyang, respectively. Just as statistical results in Table 9.4, most of low flows

fell within the 95% PIs produced by UNEEC whereas most of the high flows are above the upper prediction limit. On the other hand, quite a large number of low flows are located outside the 95% PIs estimated by the bootstrap, but some high flows are better captured by the PIs than those in the UNEEC method.

It can also be noticed that the PIs generated by the UNEEC seems to be too wide, in particular in these locations of the low flows, which leads to negative lower prediction limits. Obviously, the negative lower PL is meaningless in practice. Therefore, the accuracy of the uncertainty analysis by UNEEC should be evaluated by PICP in conjunction with other indices measuring PIs such as RMPIT. When PICP is close to the prescribed PIs, the smaller is RMPIT the better is accuracy the uncertainty prediction. It is also known that the UNEEC method is based on the model error where the point prediction model is hypothetically optimal. Hence, whether or not the optimal model is identified is an important factor affecting the reliability of accuracy of uncertainty analysis by UNEEC.

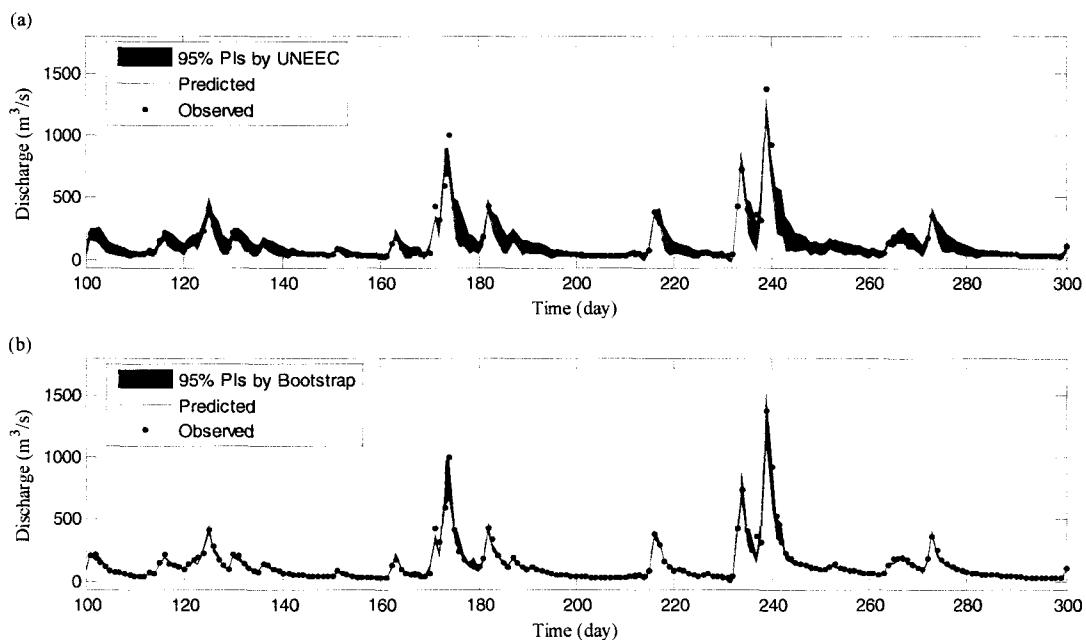


Figure 9.8 Representative details of 95% by UNEEC (a) and Bootstrap (b) for Wuxi test data using ANN model to predict one-day lead

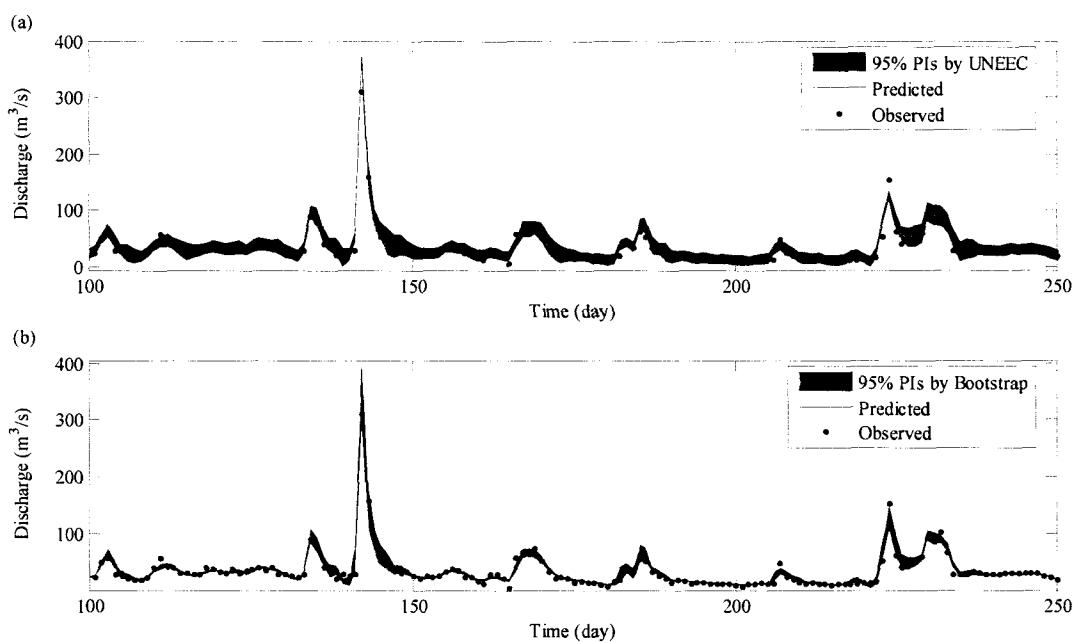


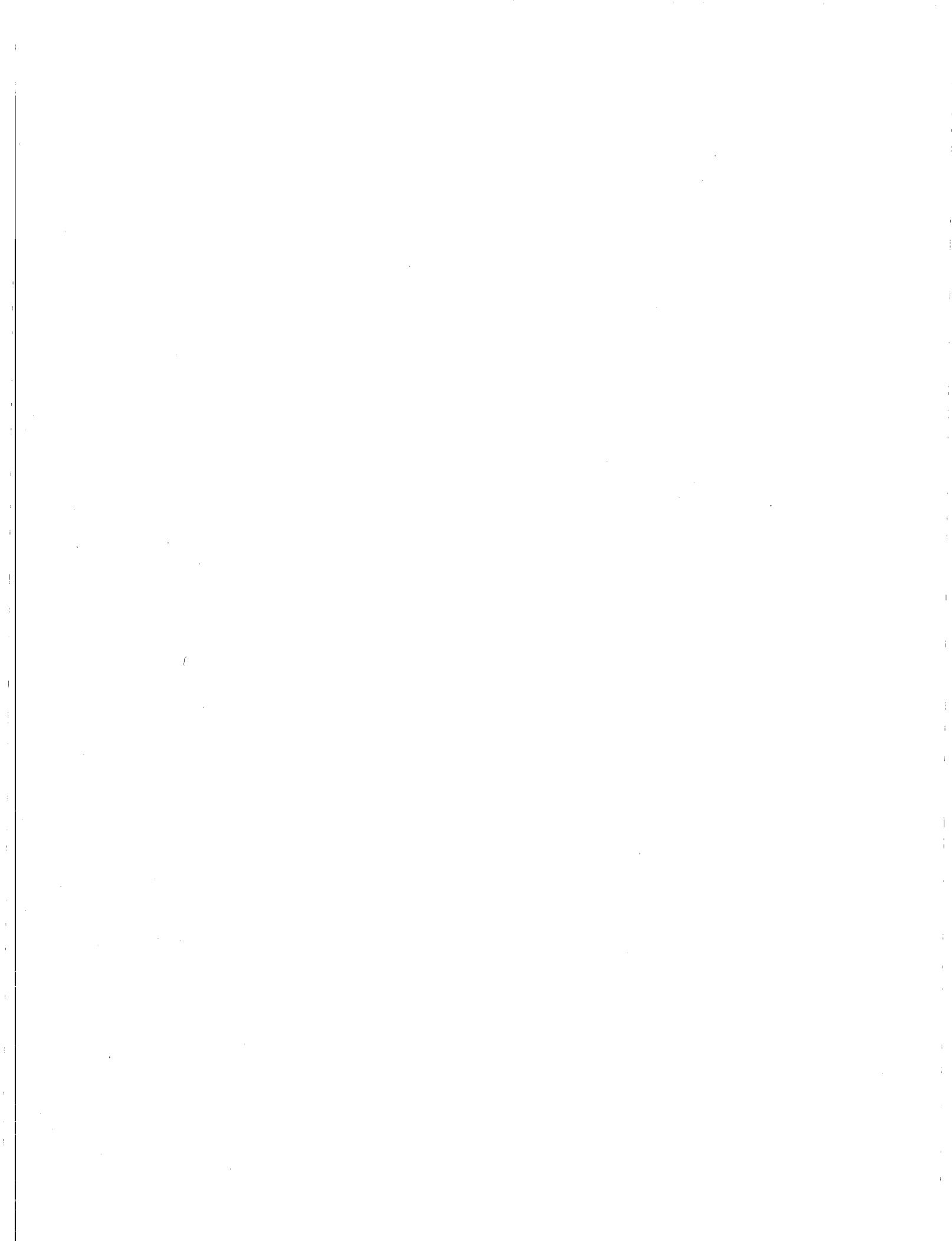
Figure 9.9 Representative details of 95% by UNEEC (a) and Bootstrap (b) for Chongyang test data using ANN model to predict one-day lead

9.4 Summary

In this chapter, point forecasts of daily flows using the ANN-SSA R-R model in Chapter 8 have been extended to interval forecasts for the uncertainty analysis purpose. The UNEEC method, which is based on the model errors, has been employed for the uncertainty analysis in comparison with the bootstrap method. UNEEC is capable of making appropriate uncertainty predictions in terms of PICP. However, some negative lower prediction limits suggest that this method may be further improved by obtaining the optimal point prediction model as much as possible. Compared with the bootstrap method where only input uncertainty is considered, UNEEC performs better in locations of low flows whereas the bootstrap method is proved to be better for the estimates of prediction intervals in locations of high flows.

Part 4

Conclusion



10 Summary and Future Work

10.1 Summary

Owing to over-simplified assumptions, inappropriate training data, model inputs, model configuration, and even individual experience of modelers, the prediction of a data-driven model tends to be full of uncertainty. This thesis is an attempt to improve the accuracy of hydrological predictions including rainfall and streamflow from three aspects: model inputs, selection of models, and data preprocessing techniques. Seven candidate methods, namely, LCA, FNN, CI, SLR, AMI, PMI, and ANN^{MOGA}, are firstly examined to select optimal model inputs in each prediction scenario. Representative models, viz., K-NN, DSBM, ANN, MANN, and ANN-SVR, are then proposed to conduct rainfall and streamflow forecasts. Four data preprocessing methods including MA, PCA, SSA and WA, are further investigated by combining them with proposed models.

K-NN, ANN, and MANN are used to predict monthly and daily rainfall series with LR as the benchmark. The comparison of seven input methods indicates that LCA is capable of reasonably identifying model inputs. In the normal mode (viz., without data preprocessing), MANN performs the best, but the advantage of MANN over ANN is not significant in monthly rainfall forecasting. Compared with results in the normal mode, the improvement of the model performance with the help of SSA is significant whereas the effect of MA or PCA on the model performance is almost negligible. In the SSA scenario, MANN also displays obvious advantages over other models, in particular for daily rainfall forecasting. In addition, two filtering approaches, supervised and unsupervised, for determining effective RCs in SSA, are evaluated. It is noticed that the unsupervised approach tends to be more effective than the supervised one. This is because the former can capture any dependence relation between model inputs and output whereas the latter is only based on the linear dependence between them.

ANN, MANN, ANN-SVR, and DSBM are employed to conduct estimates of monthly and daily streamflow series where model inputs depend only on previous flow observations and the best model inputs are also identified by LCA. In the normal mode, the global DSBM shows close performance to ANN. Compared to ANN, MANN and ANN-SVR are able to noticeably improve the accuracy of flow predictions, particularly for less smooth flow series, and they tend to be replaceable by each other. However, the prediction lag effect is observed in daily streamflow series forecasting. In the data preprocessing mode, SSA and WA are implemented in schemes, A and B. In scheme A, both SSA and WA can considerably improve the prediction accuracy and completely eradicate the prediction lag effect when they are combined with ANN, MANN and ANN-SVR. The superiority of modular models over the ANN is not significant. A comparison between SSA and WA indicates that SSA is a more effective data preprocessing technique.

ANN and MANN continue to be used to perform daily R-R prediction in which model inputs consisting of previous rainfall and streamflow observations are also indentified by LCA. Irrespective of modes, the advantage of MANN over ANN is not significant. Compared to the ANN model with only flow input, the ANN R-R model produces more accurate predictions. In the normal mode, however, the improvement of performance tends to mitigate with the increase of forecasting horizons. At one-step lead horizon, the ANN R-R model eliminates the timing error generated by the ANN model with flow input only. The situation becomes reverse in the SSA mode where the advantage of the ANN R-R model increases more significantly as the prediction horizon increases.

The above findings focus on results of point prediction, which uses the ANN-SSA R-R model. On the basis of this model, we complement this with the UNEEC method so as to attain interval prediction. The UNEEC method is then compared to the bootstrap method. Results indicate that UNEEC performs better in locations of low flows whereas the bootstrap method proves to be well suited in locations of high flows.

One of major contributions of this research is the exploration of a viable modeling

technique of coupling data-driven models with SSA. The technique has been tested with hydrological forecasting in rainfall, streamflow, and rainfall-runoff. The good agreement between predictions and observations has proved that the technique is promising. LCA has been identified as a suitable method in determining model inputs. In addition, comparison between global models (e.g. ANN) and modular models (e.g. MANN) has revealed that the advantage of modular models over global models occurs under the condition of univariate daily series prediction in the normal mode whereas the two types of models have very similar performance in the SSA mode in all prediction experiments.

10.2 Future Work

Although the findings of the study presented here have proved that the current modeling technique is promising in hydrological forecasting, there is a large amount of work to deserve further exploration.

Firstly, it can be observed that peak values are always poorly captured in both rainfall and streamflow time series predictions although the MANN-SSA model achieves the best forecasts compared to other proposed models. Therefore, new methods should be explored to improve the forecast of peak values.

Secondly, in order to conduct a more comprehensive comparison between SSA and WA, further work needs to focus on the following aspects. First, the present study employs the third order of Daubechies wavelets as the wavelet function. In general, one function can be viewed as the wavelet function if it has zero mean and be localized in both time and frequency space (Farge, 1992). Obviously, there are a large number of functions to satisfy the admissibility condition. A more appropriate wavelet function may be found for decomposition of the streamflow series. Moreover, two schemes, A and B, are adopted for the implementation of SSA or WA. Results show that scheme A is significantly superior to scheme B, in particular for SSA. As a matter of fact, scheme A is based on the unsupervised filtering method whereas scheme B is based on the supervised filtering method. To explore the potential of

scheme B, it is strongly recommended that a global search method be used to identify the optimal components for model inputs in scheme B.

Thirdly, the current study does not combine forecasted rainfall with proposed R-R models. If predicted rainfall and streamflow are used as model inputs in a dynamic multi-step forecast model, the forecasted results may be different. Moreover, the predicted rainfall is also worth being coupled with some conceptual models for multi-step forecasts.

Finally, the present proposed modeling technique should also be extended to hourly sample hydrological forecasting to ensure a wider application.

Appendix A K-Nearest-Neighbors Method

Let $\mathbf{X}(n) = [q(n), q(n-1), q(n-2)]$ be a feature vector of discharge consisting of three values of past daily recordings (viz, $m=3$) with the known number of nearest neighbors to $\mathbf{X}(n)$ (say, $K=4$). The K-NN algorithm searches through all the consecutive triplets of the historical record for the four triplets closest (in a Euclidean sense) to the present feature vector $\mathbf{X}(n)$. The predicted discharge is the mean of successors to the four closed observed feature vectors to $\mathbf{X}(n)$ as shown in Figure A.1. Hence, one-day-ahead prediction model is $\hat{q}(n+1) = \frac{1}{4} \sum_{j \in S(\mathbf{X}, n)} q(j+1)$ where $j \in [1, n]$.

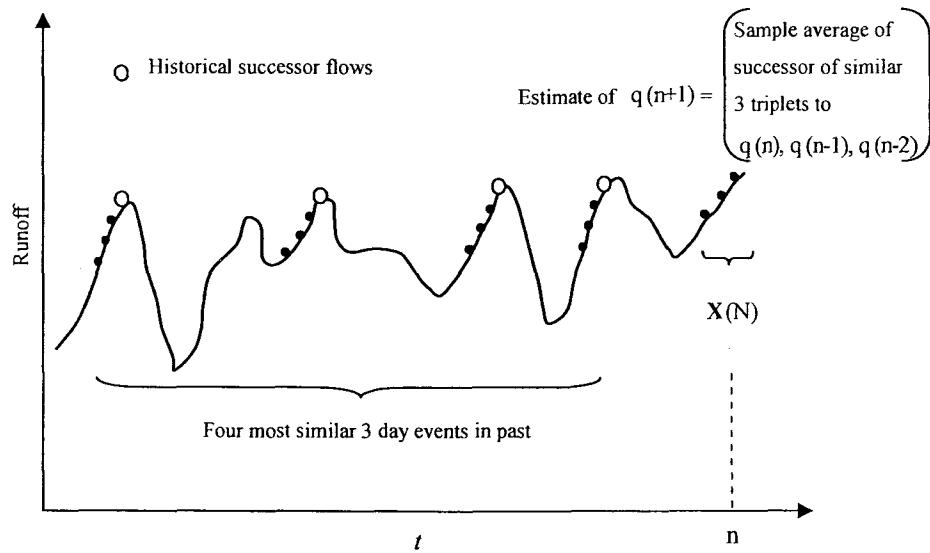


Figure A.1 K-NN method for discharge time series (adopted from Karlsson and Yakowitz, 1987a)

Appendix B Naive Model

In this model, it is assumed that the most recent past is the best indicator of the future. The method, therefore, takes the last observed value equals as the future rainfall estimate, i.e. $x_{t+T}^F = x_t$, for $\forall T$, where x_t is the observed record at instant time t , and x_{t+T}^F stands for the estimated rainfall at the lead-time T . For a modified version of the persistence model, each forecasted rainfall at the lead-time T equals to the mean value over the last T observations, given by

$$x_{t+T}^F = \sum_{i=1}^T x_{t-i+1} / T \quad (\text{B.1})$$

Appendix C Principal Component Regression

Herein, the introduction of principal component regression (PCR) is referred to Hsu et al. (2002). A multivariate linear regression model having n observations and p independent variables is given below

$$Y = X\theta + \varepsilon \quad (C.1)$$

where Y is a vector of n observations ($n \times 1$), X is $n \times p$ matrix with elements (i, j) of i th observation and j th independent variable, θ is a vector of regression coefficients, $\theta = [v_1, v_2, \dots, v_p]^T$, and ε is a vector of estimation of error ($n \times 1$) with zero mean and variance σ^2_ε . Parameters are estimated from minimizing the root mean square error of sample data and are given below

$$\hat{\theta} = (X^T X)^{-1} X^T Y \quad (C.2)$$

where $\hat{\theta}$ is the unbiased estimates of regression parameters. The above equation estimates the unbiased regression parameters that minimized the root mean square error. When the input variables are colinear, the inverse matrix of $(X^T X)^{-1}$ becomes singular, which makes finding regression parameters difficult. To reduce the uncertainty of the regression estimates, principal component transformation of input variables into uncorrelated variables before regression analyses is useful for finding more reliable regression parameters.

Substituting Eq.(4.9) for the input variable of the linear regression function in equation (D1) with, we obtain

$$Y = ZA^T\theta + \varepsilon = Z\beta + \varepsilon \quad (C.3)$$

where $\beta = A^T\theta$ are the regression parameters of principal components.

Parameters of principal component regression are estimated as

$$\hat{\beta} = (Z^T Z)^{-1} Z^T Y \quad (C.4)$$

When multicolinearities exist among original input variables, the regression parameters show high variance to those variables that are colinear to others. The

regression parameters of the original variable, θ , are given below

$$\hat{\theta} = \mathbf{A}\beta = \mathbf{A}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T\mathbf{Y} = \mathbf{A}\Lambda^{-1}\mathbf{A}^T\mathbf{X}^T\mathbf{Y} = \sum_{k=1}^p \lambda_k^{-1} e_k e_k^T \mathbf{X}^T \mathbf{Y} \quad (\text{C.5})$$

where Λ is a diagonal matrix with k th largest eigenvalue, λ_k , on k th diagonal element.

The e_k is the eigenvector of the principal component with k th largest eigenvalue.

Assume that observations are uncorrelated and have a constant variance of σ^2 for each observation y_i . The covariance matrix of $\hat{\theta}$ is given below

$$\begin{aligned} E(\hat{\theta}\hat{\theta}^T) &= \sigma^2 \mathbf{A}(\mathbf{Z}^T\mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{Z} (\mathbf{Z}^T\mathbf{Z})^{-1} \mathbf{A}^T = \sigma^2 \mathbf{A}(\mathbf{Z}^T\mathbf{Z})^{-1} \mathbf{A}^T \\ &= \sigma^2 \mathbf{A}\Lambda^{-1} \mathbf{A}^T = \sigma^2 \sum_{k=1}^p \lambda_k^{-1} e_k e_k^T \end{aligned} \quad (\text{C.6})$$

If multicollinearity appears in the original variables, \mathbf{X} , it will reveal that the eigenvalues are very small in the later principal components. The variances of the regression parameters become very large from the value of λ_k^{-1} term in the above equation. To avoid large variance on the regression parameters, those small eigenvalue terms in the calculation are removed. The new regression parameters are then expressed as

$$\tilde{\theta} = \sum_{k=1}^m \lambda_k^{-1} e_k e_k^T \mathbf{X}^T \mathbf{Y} \quad (\text{C.7})$$

The covariance of new regression parameters is reduced, and the covariance matrix of those regression parameters becomes:

$$E(\tilde{\theta}\tilde{\theta}^T) = \sigma^2 \sum_{k=1}^m \lambda_k^{-1} e_k e_k^T \quad (\text{C.8})$$

Because none of the above eigenvalues are small numbers, the variances of the estimated regression parameters are not that high. We have:

$$\theta' = \hat{\theta} - \tilde{\theta} = \sum_{k=m+1}^p \lambda_k^{-1} e_k e_k^T \mathbf{X}^T \mathbf{Y}, \quad E[\hat{\theta}] = \theta \quad (\text{C.9})$$

If the above term is nonzero, omitting this term would result in a biased estimate. However, the advantage from the reduction of parameter variance is substantial under multicollinear circumstances.

References

- Abbott, M. B., Bathurst, J. C., Cunge, J. A., O'Connell, P. E., and Rasmussen, J. (1986a), An introduction to the European Hydrologic System-Systeme Hydrologique Europeen, SHE, 1: History and philosophy of a physically-based, distributed modeling system. *Journal of Hydrology*, 87, 45-59.
- Abbott, M. B., Bathurst, J. C., Cunge, J. A., O'Connell, P. E., and Rasmussen, J. (1986b), An introduction to the European Hydrologic System-Systeme Hydrologique Europeen, SHE, 2: Structure of a physically-based, distributed modeling system. *Journal of Hydrology*, 87, 61-77.
- Abe, S. (1997), *Neural Networks and Fuzzy Systems: Theory and Applications*. Boston: Kluwer Academic Publishers.
- Abebe, A.J., and Price, R.K. (2003), Managing uncertainty in hydrological models using complementary models. *Hydrological Sciences Journal-Journal des Sciences Hydrologiques*, 48 (5), 679-692.
- Abraham, N.B., Albano, A.M., Das, B., de Guzman, G., Yong, S., Gioggia, R.S., Puccioni, G.P., and Tredicce, J.R. (1986), Calculating the dimension of attractors from small data. *Phys. Lett. A*, 114, 217-221.
- Abarbanel, H. D. I. (1996), *Analysis of Observed Chaotic Data*. Springer Verlag, New York.
- Abarbanel, H. D. I., Brown, R., Sidorowich, J. J., and Tsimring, L.S. (1993), The analysis of observed chaotic data in physical systems. *Reviews of Modern Physics*, 65(4), 1331-1392.
- Abrahart, R.J. (2003), Neural network rainfall-runoff forecasting based on continuous resampling. *Journal of Hydroinformatics*, 5(1), 51-61.
- Abrahart, R.J., Heppenstall, A.J., and See, L.M. (2007), Timing error correlation procedure applied to neural network rainfall-runoff modeling. *Hydrological Sciences Journal*, 52(3), 414-431.
- Abrahart, R.J., and See, L.M. (2000), Comparing neural network and autoregressive moving average techniques for the provision of continuous river flow forecasts in two contrasting catchments. *Hydrological Processes*, 14, 2157-2172.
- Abrahart, R.J. and See, L.M. (2002), Multi-model data fusion for river flow forecasting: an evaluation of six alternative methods based on two contrasting catchments. *Hydrology and Earth System Sciences*, 6(4), 655-670.
- Abrahart, R. J., and See, L. M. (2007), Neural network modelling of non-linear hydrological relationships. *Hydrology and Earth System Sciences*, 11(5), 1563-1579.
- Abrahart, R.J., See, L.M., and Kneale, P.E. (1999), Using pruning algorithms and genetic algorithms to optimise network architectures and forecasting inputs in a neural network rainfall-runoff model. *Journal of Hydroinformatics*, 1, 103-114.

- Abrahart, R.J., See, L.M., and Kneale, P.E. (2001), Applying saliency analysis to neural network rainfall-runoff modelling. *Computers and Geosciences*, 27, 921-928.
- Adarnowski, J.F. (2008), Development of a short-term river flood forecasting method for snowmelt driven floods based on wavelet and cross-wavelet analysis. *Journal of Hydrology*, 353(3-4), 247-266.
- Aha, D., Kibler, D., and Albert, M. (1991), Instance-based learning algorithms. *Machine Learning*, 6, 37-66.
- Ahmad, S., and Simonovic, S.P. (2005), An artificial neural network model for generating hydrograph from hydro-meteorological parameters. *Journal of Hydrology*, 315, 236-251.
- Al-Alawi, S.M., Abdul-Wahab, S. A., and Bakheit, C. S. (2008), Combining principal component regression and artificial neural networks for more accurate predictions of ground-level ozone. *Environmental Modelling and Software*, 23, 396-403.
- Alvisi, S., Mascellani, G., Franchini, M., Bárdossy, A. (2006), Water level forecasting through fuzzy logic and artificial neural network approaches. *Hydrology and Earth System Sciences*, 10, 1-17.
- Amari, S.I., Murata, N., Müller, K.-R., Finke, M., and Yang, H.H. (1997), Asymptotic statistical theory of overtraining and cross-validation. *IEEE Transactions on Neural Networks* 8 (5), 985-996.
- Anctil, F., and Lauzon, N. (2004), Generalization for neural networks through data sampling and training procedures, with applications to streamflow predictions. *Hydrology and Earth System Sciences*, 8(5), 940-958.
- Anctil, F., Perrin, C., and Andre'assian, V. (2003), ANN output updating of lumped conceptual rainfall/runoff forecasting models. *Journal of the American Water Resources Association*, 39(5), 1269-1279.
- Anctil, F., Perrin, C., and Andréassian, V. (2004), Impact of the length of observed records on the performance of ANN and of conceptual parsimonious rainfall-runoff forecasting models. *Environmental Modeling and Software*, 19, 357-368.
- Anders, U., and Korn, O. (1999), Model selection in neural networks. *Neural Networks*, 12, 309–323.
- Aqil, M., Kita, I., Yano, A., and Nishiyama, S. (2006), Prediction of flood abnormalities for improved public safety using a modified adaptive neuro-fuzzy inference system. *Water Science & Technology*, 54(11-12), 11-19.
- Aqil, M., Kita, I., Yano, A., and 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.
- Arduino, G., Reggiani, P., and Todini, E. (2005), Recent advances in flood forecasting and flood risk assessment. *Hydrology and Earth System Sciences*, 9(4), 280-284.

- ASCE. (2000a), Artificial neural networks in hydrology 1: Preliminary concepts. *Journal of Hydrologic Engineering*, 5(2), 115-123.
- ASCE. (2000b), Artificial neural networks in hydrology 2: Hydrology applications. *Journal of Hydrologic Engineering*, 5(2), 124-137.
- Babovic, V. and Abbott, M. B. (1997), The evolution of equations from hydraulic data, Part II: Applications. *Journal of Hydraulic Resources*, 35, 411-430.
- Babovic, V., Cañizares, R., Jensen, H.R. and Klinting, A. (2001), Neural networks as routine for error updating of numerical models. *Journal of Hydraulic Engineering*, 127(3), 181-193.
- Babovic, V., Harris, E. and Falconer, R. (2003), Velocity predictions in compound channels with vegetated floodplains using genetic programming. *International Journal of River Basin Management*, 2 (1), 117-125.
- Babovic, V., and Keijzer, M. (2000), Genetic programming as a model induction engine, *Journal of Hydroinformatics*, 2(1), 35- 61.
- Babovic, V., and Keijzer, M. (2002), Rainfall runoff modelling based on genetic programming. *Nordic Hydrol*, 33 (5), 331-346.
- Babovic, V. and Keijzer, M. (2005), Rainfall runoff modelling based on genetic programming. In *Encyclopedia of Hydrological Sciences*, vol 1. (ed. Andersen, M.G.). John Wiley & Sons, New York, Doi: 10.1002/0470848944.hsa017.
- Babovic, V., Keijzer, M. and Stefansson, M. (2000), Optimal embedding using evolutionary algorithms. In: Proc. 4th Int. Conference on Hydroinformatics, Cedar Rapids.
- Bagis A., and Karaboga, D. (2004), Artificial neural networks and fuzzy logic based control of spillway gates of dams. *Hydrological Processes*, 18(13), 2485-2501.
- Bannayan, M., and Hoogenboom, G. (2008), Predicting realizations of daily weather data for climate forecasts using the non-parametric nearest-neighbor re-sampling technique. *International Journal of Climatology*, 28(10), 1357-1368.
- Baratta et al., Baratta, D., Cicioni, G., Masulli, F. and Studer, L. (2003), Application of an ensemble technique based on singular spectrum analysis to daily rainfall forecasting. *Neural Networks*, 16, 375-387.
- Battiti, R., (1994), Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5 (4), 537-550.
- Benítez, J. M., Castro, J. L., and Requena, I. (1997), Are Artificial Neural Networks Black Boxes? *IEEE Transactions on Neural Networks*, 8(5), 1156-1164.
- Bergstrom, S. (1995), Chapter 13: The HBV model. Computer models of watershed hydrology, V. P. Singh, ed., Water Resources Publications, Littleton, Colo.
- Beven, K. J. (1995), Chapter 18: TOPMODEL. Computer models of watershed hydrology, V. P. Singh, ed., Water Resources Publications, Littleton, Colo.

- Beven K. (1993), Prophecy, reality and uncertainty in distributed hydrological modelling. *Advances in Water Resources*, 16, 41-51.
- Beven, K.J. (2000), Rainfall-runoff modeling: The primer. John Wiley & Sons, Chichester, U.K.
- Beven, K, and Binley, A. (1992), The future of distributed models: model calibration and uncertainty prediction. *Hydrological Processes*, 6, 279-298.
- Beven, K. J., Calver, A., and Morris, E. (1987), The Institute of Hydrology distributed model. *Institute of Hydrology Rep. No. 98*, Wallingford, U.K.
- Bezdek, J.C. (1981), Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York.
- Birkundavyi, S., Labib, R., Trung, H.T., and Rousselle, J. (2002), Performance of neural networks in daily streamflow forecasting. *Journal of Hydrologic Engineering*, 7(5), 392-398.
- Box, G.E., and Jenkins, G.M. (1976), Time Series Analysis: Forecasting and Control. Revised Edition, Holden-Day, San Francisco, California.
- Box, G.E.P., Jenkins, G.M., and Reinsel, G.C., (1994), Time Series Analysis: Forecasting and Control, 3rd edition, Prentice Hall.
- Bowden, G.J., Dandy, G.C., and Maier, H.R. (2005), Input determination for neural network models in water resources applications: Part 1—background and methodology. *Journal of Hydrology*, 301, 75-92.
- Bowerman, B. L. and O'Connell, R.T. (1987), Time series forecasting, unified concepts and computer implementation, 2nd ed. Duxbury Press: Boston.
- Brath, A., Montanari, A., and Toth, E. (2002), Neural networks and non-parametric methods for improving realtime flood forecasting through conceptual hydrological models. *Hydrology and Earth System Sciences*, 6(4), 627-640.
- Bray, M. and Han, D. (2004), Identification of support vector machines for runoff modelling. *Journal of Hydroinformatics*, 6(4), 265-280.
- Breiman, L. (1994), Stacked regressions, Technical Report No. 367, Department of statistics, University of California at Berkeley, USA.
- Burden, F.R., Brereton, R.G., and Walsh, P.T., (1997), Cross-validatory selection of test and validation sets in multivariate calibration and neural networks as applied to spectroscopy. *Analyst* 122 (10), 1015-1022.
- Burke, L.I., and Ignizio, J.P., (1992), Neural networks and operations research: an overview. *Computer and Operations Research*, 19 (3/4), 179–189.
- Campolo, M., Andreussi, P., and Soldati, A. (1999), River flood forecasting with a neural network model, *Water Resources Research*, 35 (4), 1191-1197.

- Campolo, M., Andreussi, P. and Soldati, A. (2003), Artificial neural network approach to flood forecasting in the river Arno. *Hydrological Sciences Journal*, 48(3), 381-398.
- Cannas, B., Fanni, A., Pintus, M., and Sechi, G. M. (2002), Neural Network Models to Forecast Hydrological Risk. *IEEE IJCNN*, 2002.
- Cannas, B., Fanni, A., See, L., and Sias, G. (2006), Data preprocessing for river flow forecasting using neural networks: Wavelet transforms and data partitioning. *Physics and Chemistry of the Earth*, 31, 1164-1171.
- Cannon, A. J. and Whitfield, P. H. (2002), Downscaling recent streamflow conditions in British Columbia, Canada, using ensemble neural network models. *Journal of Hydrology*, 259, 136-151.
- Carpenter, W. C., and Barthelemy, J. (1994), Common misconcepts about neural networks as approximators. *Journal of Computing in Civil Engineering, ASCE*, 8(3), 345-358.
- Calver, A., and Wood, W. L. (1995), Chapter 17: The Institute of Hydrology distributed model. *Computer models of watershed hydrology*, V. P. Singh, ed., Water Resources Publications, Littleton, Colo.
- Cannon, A.J., and McKendry, I.G. (2002), A graphical sensitivity analysis for statistical climate models: Application to Indian monsoon rainfall prediction by artificial neural networks and multiple linear regression models. *International Journal of Climatology*, 22 (13), 1687-1708.
- Carlson, R.F., McCormick, J.A. and Watts, D.G. (1970), Application of linear random models to four annual stream-flows series. *Water Resources Research*, 6, 1070–1078.
- Carvajal L.F., Salazar, J.E., Mesa, O.J., and Poveda, G. (1998), Hydrological prediction in Colombia using singular spectral analysis and the maximum entropy method. *Ingenieria Hidraulica En Mexico*, 13(1), 7-16.
- Castellano-Me'ndeza, M., Gonza'lez-Manteigaa, W., Febrero-Bande, M., Manuel Prada-Sa'ncheza, J., and Lozano-Caldero'n, R. (2004), Modeling of the monthly and daily behavior of the runoff of the Xallas river using Box-Jenkins and neural networks methods. *Journal of Hydrology*, 296, 38-58.
- Chan, J.C.L., and Shi, J.E. (1999), Prediction of the summer monsoon rainfall over South China. *International Journal of Climatology*, 19 (11), 1255-1265.
- Chang, L.C., and Chang, F.J. (2001), Intelligent control of modeling of real time reservoir operation. *Hydrological Processes*, 15, 1621-1634.
- Chang, F.J., Chang, L.C., and Huang, H.L. (2002), Real-time recurrent learning neural network for stream-flow forecasting. *Hydrological Processes*, 16, 2577-2588.
- Chang, L. C., Chang, F. J., and Tsai, Y. H. (2005), Fuzzy exemplar-based inference system for flood forecasting. *Water Resources Research*, 41, W02005, doi:10.1029/2004WR003037.

Chang, F. J. and Chen, L. (1998), Real-coded genetic algorithm for rule-based flood control reservoir management. *Water Resource Management*, 12(3), 185-198.

Chang, F.J. and Chen, Y.C. (2001), A Counter propagation Fuzzy-Neural Network Modeling Approach to Real-time Stream flow Prediction. *Journal of Hydrology*, 245, 153-164.

Chang, C.H., and Wu, Y.C. (1995), Genetic algorithm based tuning method for symmetric membership functions of fuzzy logic control system. In Proc. IEEE/IAS International Conference on Industrial Automation and Control: Emerging Technologies, 421-428, Taipei.

Chattopadhyay,S., and Chattopadhyay, G. (2007), Identification of the best hidden layer size for three-layered neural net in predicting monsoon rainfall in India. *Journal of Hydroinformatics*, 10(2), 181-188.

Chau, K.W. (2006), Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River, *Journal of Hydrology*, 329 (3-4), 363-367.

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

Chetan, M., and Sudheer K. P. (2006), A hybrid linear-neural model for river flow forecasting. *Water Resources Research*, Vol.42, W04402, doi: 10.1029/2005WR004072, 2006.

Chen, S.H., Lin, Y.H., Chang, L.C., and Chang F. J. (2006), The Strategy of Building a Flood Forecast Model by Neuro-Fuzzy Network. *Hydrological Processes*, 20, 1525-1540.

Chen, S.Y., and Yu, G. (2006), Variable fuzzy sets and its application in comprehensive risk evaluation for flood-control engineering system. *Fuzzy Optimization and Decision Making*, 5(2), 153-162.

Cheng, B., and Titterington, D. M. (1994), Titterington Neural networks: A review from a statistical perspective. *Statistical Science*, 9(1), 2-54.

Cheng, J., Qian, J.S., and Guo, Y.N. (2006), A Distributed Support Vector Machines Architecture for Chaotic Time Series Prediction. Irwin King, Jun Wang, Laiwan Chan, DeLiang L. Wang (Eds.): *Neural Information Processing*, 13th International Conference, ICONIP 2006, Hong Kong, China, October 3-6, 2006, Proceedings, Part I. *Lecture Notes in Computer Science*, 4232, 892-899.

Cherkassky, V. and Ma, Y. (2004), Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, 17(1), 113-126.

Cherkassky, V. and Mulier, F. (1998), *Learning From Data: Concepts, Theory and Methods*. Wiley, New York.

Chetan, M., and Sudheer K. P. (2006), A hybrid linear-neural model for river flow forecasting. *Water Resources Research*, Vol.42, W04402, doi: 10.1029/2005WR004072, 2006.

- Chibanga, R., Berlamont, J., and Vandewalle, J. (2003), Modelling and forecasting of hydrological variables using artificial neural networks: the Kafue River sub-basin. *Hydrological Sciences Journal*, 48(3), 363-379.
- Chiu, S.L. (1994), Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2, 267-278.
- Chng, E.S., Chen, S., Mulgrew, B., (1996), Gradient radial basis function networks for nonlinear and nonstationary time series prediction. *IEEE Transactions on Neural Networks* 7 (1), 191-194.
- Chryssolouris, G., Lee, M., and Ramsey, A. (1996), Confidence interval prediction for neural network models. *IEEE Transactions on Neural Networks*, 7(1), 229-232.
- Chon, K.H., and Cohen, R.J. (1997), Linear and nonlinear ARMA model parameter estimation using an artificial neural network. *IEEE Transactions on Biomedical Engineering* 44 (3), 168-174.
- Chow, T.W.S., and Huang, D., (2005), Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information. *IEEE Transactions on Neural Networks* 16 (1), 213-224.
- Chryssolouris, G., Lee, M., and Ramsy, A. (1996), Confidence interval prediction for neural-network models. *IEEE Transactions on Neural Networks*, 7, 229-236.
- Chu, P.S., and He, Y.X. (1994), Long-Range Prediction of Hawaiian Winter Rainfall Using Canonical Correlation-Analysis). *International Journal of Climatology*, 14(6), 659-669.
- Civanlar, M.R., and Trussell, H.J. (1986), Constructing membership functions using statistical data. *Fuzzy Sets and Systems*, 18, 1-13.
- Clair, G.J., Clair, K. U. S., and Yuan, B. (1997), *Fuzzy Set Theory: Foundations and Applications*. Prentice-Hall, Inc.
- Corani, G., and Guariso, G. (2005), Coupling Fuzzy Modeling and Neural Networks for River Flood Prediction. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(3), 382-390.
- Cordón, O., Herrera, F., Hoffmann, F., and Magdalena, L. (2001), *Genetic Fuzzy Systems Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. World Scinetific.
- Corzo, G., and Solomatine, D. (2007), Baseflow separation techniques for modular artificial neural network modelling in flow forecasting. *Hydrological Sciences–Journal–des Sciences Hydrologiques*, 52(3), 491-507.
- Coulibaly, P., Anctil, F., and Bobée, B. (2000), Daily reservoir inflow forecasting using artificial neural networks with stopped training approach *Journal of Hydrology*, 230, 244-257.

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

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

Darbellay, G.A., (1999), An estimator of the mutual information based on a criterion for independence. *Computational Statistics and Data Analysis*, 32, 1-17.

Daubechies, I., (1992), Ten Lectures on Wavelets CSBM-NSF Series. *Application Mathematics*, 61. SIAM publication, Philadelphia PA.

Davidson, J. W., Savic, D. A. and Walters, G. A. (1999), Method for Identification of explicit polynomial formulae for the friction in turbulent pipe flow. *Journal of Hydroinformatics*, 2(1), 115-126.

Davidson, J.W., Savic, D. A. and Walters, G. A. (2000), Approximators for the Colebrook-White formula obtained through a hybrid regression method. In *Computational Methods in Water Resources Vol 2 Computational Methods, Surface Water Systems and Hydrology* (ed. L. R. Bentley, J. F. Sykes, C. A. Brebbia, W. G. Gray, and G. F. Pinder), pp. 983–989. Balkema, Rotterdam.

Davolio, S., Miglietta, M. M., Diomede, T., Marsigli, C., Morgillo, A., and Moscatello, A. (2008), A meteo-hydrological prediction system based on a multi-model approach for precipitation forecasting. *Natural hazards and earth system sciences*, 8 (1), 143-159.

Dawson, C. W. and Wilby, R. L. (1999), A comparison of artificial neural networks used for river flow forecasting, *Hydrology and Earth System Sciences*, 3, 529–540.

Dawson, C.W., and Wilby, R.L. (2001), Hydrological Modeling Using Artificial Neural Networks. *Progress in Physical Geography*, 25(1), 80-108.

Deka, P. and Chandramouli,V. (2003), A fuzzy neural network model for deriving the river stage-discharge relationship. *Hydrological Sciences Journal*, 48(2), 197-210.

Deka, P., and Chandramouli, V. (2005), Fuzzy Neural Network Model for Hydrologic Flow Routing. *Journal of Hydrologic Engineering*, 10(4), 302-314.

DelSole, T., and Shukla, J. (2002), Linear prediction of Indian monsoon rainfall. *Journal of Climate*, 15 (24), 3645-3658.

Deo, M.C., and Thirumalaiah, K. (2000), Real time forecasting using neural networks. In *artificial neural networks in hydrology*, (eds.) Govindaraju, R.S., and Rao, A.R. Water Science and Technology Library.

Despic, O. and Simonovic, S P. (2000), Aggregation operators for soft decision making in water resources. *Fuzzy Sets and Systems*, 115, 11-13.

- de Vos, N.J. and Rientjes, T.H.M. (2005), Constraints of artificial neural networks for rainfall -runoff modeling: trade-offs in hydrological state representation and model evaluation. *Hydrology and Earth System Sciences*, 9, 111-126.
- de Vos, N. J. and Rientjes, T. H. M. (2007), Correction of timing errors of artificial neural network rainfall-runoff models. In: *Hydroinformatics in Practice: Computational Intelligence and Technological Developments in Water Applications* (ed. by R. J. Abrahart, L. M. See & D. Solomatine). Springer.
- de Vos, N.J., and Rientjes, T.H.M. (2007), Multi-objective performance comparison of an artificial neural network and a conceptual rainfall-runoff model. *Hydrological Sciences Journal-Journal Des Sciences Hydrologiques*, 52 (3), 397-413.
- Dibike, Y. B. and Solomatine, D. P. (2001), River flow forecasting using artificial neural networks, *Physics and Chemistry of the Earth (B)*, 26 (1), 1-7, 2001.
- Dibike, Y. B., Velickov, S., Solomatine, D., and Abbott, M. B. (2001), Model induction with support vector machines: introduction and applications. *Journal of Computing in Civil Engineering*, 15(3), 208-216.
- Diomede, T., Davolio, S., Marsigli, C., Miglietta, M. M., Moscatello, A., Papetti, P., Paccagnella, T., Buzzi, A., and Malguzzi, P. (2008), Discharge prediction based on multi-model precipitation forecasts. *Meteorology and Atmospheric Physics*, 101 (3-4), 245-265.
- Doering, A., Galicki, M., and Witte, H., (1997), Structure optimization of neural networks with the A*-algorithm. *IEEE Transactions on Neural Networks*, 8(6), 1434-1445.
- Draper, N. R. and Smith, H. (1998), *Applied regression analysis*, 3rd ed. New York: Wiley.
- Duband, D., Obled, Ch. and Rodriguez, J. Y. (1993), Unit hydrograph revisited: an alternate iterative approach to UH and effective precipitation identification. *Journal of Hydrology*, 150(1), 115-149.
- Dubrovin, T., Jolma, A., and Turunen, E. (2002), Fuzzy Model for Real-Time Reservoir Operation. *Journa of Water Resources Planning and Management*, 128(1), 66-73.
- Elsner, J., and Tsonis, A. (1996), *Singular Spectrum Analysis. A New Tool in Time Series Analysis*. New York: Plenum Press.
- Elshorbagy, A., Simonovic, S.P., and Panu, U.S. (2002), Estimation of missing stream flow data using principles of chaos theory. *Journal of Hydrology*, 255, 123–133.
- Essex, C., and Nerenberg, M.A.H. (1991), *Proc. R. Soc. London, Series A*, 435, 287.
- Faraway, J., and Chatfield, C. (1998), Time series forecasting with neural networks: a comparative study using the airline data. *Applied Statistics*, 47 (2), 231-250.

- Farge, M., (1992), Wavelet transforms and their applications to turbulence. Annual Review of Fluid Mechanics, 24, 395-457.
- Farmer, J. D., and Sidorowich, J. J. (1987), Predicting chaotic time series, Physical Review Letter, 59(4), 845-848.
- Fraser, A.M., and Swinney, H.L. (1986), Independent coordinates for strange attractors from mutual information. Physical Review A, 33(2), 1134-1140.
- Fernando, D.A.K., and Jayawardena, A.W. (1998), Runoff forecasting using RBF networks with OLS algorithm. Journal of Hydrologic Engineering, 3(3), 203-209.
- Fernando, T.M.K.G., Maier, H.R. and Dandy, G.C. (2009), Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach. Journal of Hydrology, 367, 165-176.
- Fletcher, R. (1987), Practical Methods of Optimization (second edn). Wiley & Sons Inc., New York, USA.
- Fortin, V., Ouarda, T.B.M.J., and Bobe'e, B., (1997), Comment on 'The use of artificial neural networks for the prediction of water quality parameters' by H.R. Maier and G.C. Dandy. Water Resources Research, 33 (10), 2423-2242.
- Fraser, A.M. and Swinney, H.L. (1986), Independent coordinates for strange attractors from mutual information, Physical Review A, 33(2), 1134-1140.
- French, M.N., Krajewski, W.F., and Cuykendall, R.R. (1992), Rainfall forecasting in space and time using a neural network. Journal of Hydrology, 137, 1-31.
- Galeati, G. (1990), A comparison of parametric and non-parametric methods for runoff forecasting. Hydrological Science Journal, 35(1), 79-84.
- Ganguly, A.R., and Bras, R.L. (2003), Distributed quantitative precipitation forecasting (DQPF) using information from radar and numerical weather prediction models. Journal of Hydrometeorology, 4 (6), 1168-1180.
- García-Pedrajas, N., Ortiz-Boyer, D., and Hervás, C. (2006), An alternative approach for neural network evolution with a genetic algorithm: Crossover by combinatorial optimization. Neural Network, 19, 514-528.
- Gaume, E. and Gosset, R. (2003), Over-parameterisation, a major obstacle to the use of artificial neural networks in hydrology? Hydrology and Earth System Sciences, 7, 693-706.
- Gevrey, M., Diomopoulos, I., and Lek, S. (2003), Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecological Modeling, 160(3), 249-264.
- Gevrey, M., Dimopoulosb, I., and Lek, S. (2006), Two-way interaction of input variables in the sensitivity analysis of neural network models. Ecological modelling, 195, 43-50.

- Ghilardi, P., and Rosso, R. (1990), Comment on chaos in rainfall, *Water Resources Research*, 26 (8), 1837-1839.
- Giustolisi, O. and Laucelli, D. (2005), Improving generalization of artificial neural networks in rainfall-runoff modeling. *Hydrological Sciences–Journal–des Sciences Hydrologiques*, 50(3), 439-457.
- Giustolisi, O., and Savic, D. A. (2006), A symbolic data-driven technique based on evolutionary polynomial regression, *Journal of Hydroinformatics*, 8(3), 207-222.
- Giustolisi, O. and Simeone, V. (2006), Optimal design of artificial neural networks by multi-objective strategy: groundwater level predictions. *Hydrologic Science Journal*, 51(3), 502-523.
- Golyandina, N. Nekrutkin, V., and Zhigljavsky, A. (2001), *Analysis of Time Series Structure: SSA and Related Techniques*, Chapman & Hall/CRC.
- Grassberger, P., and Procaccia, I. (1983), Measuring the strangeness of strange attractors. *Physica* 9D, 189-208.
- Guhathakurta, P. (2008), Long lead monsoon rainfall prediction for meteorological sub-divisions of India using deterministic artificial neural network model. *Meteorology and Atmospheric Physics*, 101 (1-2), 93-108.
- Guhathakurta, P., Rajeevan, M., and Thapliyal, V. (1999), Long range forecasting Indian summer monsoon rainfall by a hybrid Principal Component neural network model. *Meteorology and atmospheric physics*, 71, (3-4), 255-266.
- Gunn, S.R. (1998), Support vector machines for classification and regression. *Image, Speech and Intelligent Systems Tech. Rep.*, University of Southampton, U.K.
- Gustafson, D., and Kessel, W. (1979), Fuzzy clustering with a fuzzy covariance matrix. In *Proceedings of IEEE CDC 1979*, San Diego, CA, USA, 761-766.
- Gupta, H. V., Beven, K. J., and Wagener, T. (2005), Model calibration and uncertainty estimation, in *Encyclopedia of Hydrological Sciences*, edited by M. G. Anderson, pp. 2015- 2031, John Wiley, New York.
- Haltiner, J.P. and Salas, J.D. (1988), Development and testing of a multivariate seasonal ARMA(1,1) model. *Journal of Hydrology*, 104, 247-272.
- Han, D., Cluckie, I. D., Karbassioun, D., Lawry, J. and Krauskopf, B. (2002), River Flow Modelling Using Fuzzy Decision Trees. *Water Resources Management*, 16, 431-445.
- Han, D.W., Kwong, T., and Li, S. (2007), Uncertainties in real-time flood forecasting with neural networks. *Hydrological Processes*, 21, 223-228.
- Hansen, J.V., Nelson, R.D. (1997), Neural networks and traditional time series methods: A synergistic combination in state economic forecasts. *IEEE Transactions on Neural Networks*, 8 (4), 863–873.

Hasebe, M., Kumekawa, T., and Sato, T. (1992), The application to the dam gate operation rule from a viewpoint of water resources by using various reasoning method of fuzzy set theory. Applications of Artificial Intelligence in Engineering, 7, 561-578.

Hasebe, M., and Nagayama, Y. (2002), Reservoir operation using the neural network and fuzzy systems for dam control and operation support. Advances in Engineering Software, 33(5): 245-260.

Hasebe, M., Kumekawa, T. and Kurosaki, M. (1998), Further study of control system for reservoir operation aided by neural networks and fuzzy set theory. Transaction on Information and Communication Technologies, 20.

Heskes, T. (1997), Practical confidence and prediction interval. In M. Mozer, et al. (Ed.), Advances in neural information processing systems: Vol. 9 (pp.176-182). Cambridge, MA: MIT Press.

Hill, T., Marquez, L., O'Connor, M., Remus, W., (1994), Artificial neural network models for forecasting and decision making. International Journal of Forecasting, 10, 5-15.

Homaifar, A., and Mc Connick, E. (1995), Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms. IEEE Trans. fuzzy Systems, 3(2), 129-139.

Hornik, K., Stinchcombe, M., and White, H. (1989), Multilayer feedforward networks are universal approximators. Neural Networks, 2, 359–366.

Hong, S.Z., and Hong, S.M., (1994), An amendment to the fundamental limits on dimension calculations. Fractals, 2 (1), 123-125.

Hotelling, H., (1933), Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24, 417-441.

Hsu, K.L., Gupta, H.V., Gao, X.G., Sorooshian, S., and Imam, B. (2002), Self-organizing linear output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis. Water Resources Research, 38(12), 1302,doi:10.1029/2001WR000795.

Hsu, K.L., Gupta, H.V., and Sorooshian, S. (1995), Artificial neural network modeling of the rainfall-runoff process. Water Resources Research, 31(10), 2517-2530.

Hsu, C. W., Chang, C.C. and Lin, C. J. (2003), A practical guide to support vector classification. Available at <http://www.csie.ntu.edu.tw/cjlin/papers/guide/guide.pdf>. [Accessed on 20/6/07].

Hu, T.S., Lam, K.C., Ng, S.T., (2001), River flow time series prediction with a range-dependent neural network. Hydrological Science Journal, 46 (5), 729-745.

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

- Huang W.R., Xu B., and Hilton, A. (2004), Forecasting Flows in Apalachicola River Using Neural Networks. *Hydrological Processes*, 18, 2545-2564.
- Hundecha, Y., Bardossy, A., and Theisen, H.W. (2001), Development of a fuzzy logic-based rainfall-runoff model. *Hydrological Sciences Journal*, 46, 363-376.
- Ichiyanagi, K., Goto, Y., Mizuno, K., Yokomizu, Y., and Matsumura, T. (1995), An Artificial Neural Network to Predict River Flow Rate into a Dam for a Hydro-Power Plant. Proc. Of 1995 IEEE Int. Conf. on Neural Networks (Perth), 5, 2679-2682.
- Imrie, C.E, Durucan, S, and Korre, A. (2000), River flow prediction using artificial neural networks: generalization beyond the calibration range. *Journal of Hydrology*, 233, 138-153.
- Islam, M.N., Liong, S.Y., Phoon, K.K., and Liaw, C.Y. (2001), Forecasting of river flow data with a general regression neural network. *Integrated water resources management(Proceedings of a symposium held at Davis, California, April 2000)*, IAHS Publ. no. 272.
- Jacquin, A. P., and Shamseldin, A. Y. (2004), Development of a river flow forecasting model for the Blue Nile River using Takagi-Sugeno-Kang fuzzy systems. In: *Hydrology: Science and Practice for the 21st Century, Proceedings of the British Hydrological Society International Conference, Imperial College, London, July Vol. I*, British Hydrological Society, London. pp. 291-296.
- Jacquin, A. P., and Shamseldin, A.Y. (2006), Development of rainfall-runoff models using Takagi-Sugeno fuzzy inference systems. *Journal of Hydrology*, 329(1-2), 154-173.
- Jacquin, A. P., and Shamseldin, A.Y. (2009), Review of the applications of fuzzy inference systems in river flow forecasting. *Journal of Hydroinformatics*, 11(3-4), 202-210.
- Jain, S. K., Das, A., and Drivastava, D. K. (1999), Application of ANN for reservoir inflow prediction and operation. *Journal of Water Resources Planning and Management*, 125(5), 263-271.
- Jain, A., and Srinivasulu, S. (2004), Development of effective and efficient rainfall-runoff models using integration of deterministic, real-coded genetic algorithms and artificial neural network techniques, *Water Resource Research*, 40, W04302.
- Jain, A., and Srinivasulu, S. (2006), Integrated approach to model decomposed flow hydrograph using artificial neural network and conceptual techniques. *Journal of Hydrology*, 317, 291-306.
- Jain, A., Sudheer, K.P., and Srinivasulu, S. (2004), Identification of physical processes inherent in artificial neural network rainfall runoff models. *Hydrological Processes*, 18 (3), 571-581.
- Jang, J. S. R. (1993), ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(3), 665–685.

Jayawardena, A.W., and Fernando, D.A.K. (1998), Use of radial basis function type artificial neural networks for runoff simulation. Computer-Aided Civil and Infrastructure Engineering, 13 (2), 91-99.

Jayawardena, A.W., and Gurung, A.B. (2000), Noise reduction and prediction of hydrometeorological time series: dynamical systems approach vs. stochastic approach. Journal of hydrology, 228, 242-264.

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

Jeong, D.I., and Kim, Y.O. (2005), Rainfall-runoff models using artificial neural networks for ensemble streamflow prediction. Hydrological Processes, 19 (19), 3819-3835.

Jolliffe, I.T. (2002), Principal Component Analysis, 2nd ed. Publisher: Springer.

Kachroo, R.K. (1992a), River flow forecasting: Part 1- A discussion of the principles. Journal of Hydrology, 133, 1-15.

Kachroo, R.K. (1992b), River flow forecasting: Part 5- Applications of a conceptual model. Journal of Hydrology, 133, 141-178.

Kantz, H., and Schreiber, T. (2004), Nonlinear time series analysis (2nd edition). Springer.

Karaboga, D., Bagis, A., and Haktanir, T. (2004), Fuzzy logic based operation of spillway gates of reservoirs during floods. Journal of Hydrological Engineering, 9(6), 544-549.

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

Karlsson, M., and Yakowitz, S., (1987b), Rainfall-runoff forecasting methods, old and new. Stochastic Hydrology and Hydraulics, 1, 303-318.

Karr, C.L. (1991a), Applying genetic to fuzzy control. IEEE AI Exper, 11(6), 26-33.

Karr, C.L. (1991b), Design of an adaptive fuzzy logic controller using a genetic algorithm. Proc.of 4th Int.Conf.on Genetic Algorithms, 450-457.

Kecman, V. (2001), Learning and soft computing: support vector machines, neural networks, and fuzzy logic models. MIT press, Cambridge, Massachusetts.

Kember, G., and Flower, A.C. (1993), Forecasting river flow using nonlinear dynamics. Stochastic Hydrology and Hydraulics, 7, 205-212.

Kennel, M. B., Brown, R., and Abarbanel, H. D. I. (1992), Determining embedding dimension for phase space reconstruction using geometrical construction. Physical Review A., 45(6), 3403-3411.

- Keskin, M. E., Taylan, D., and Terzi, Ö. (2005), Adaptive neural-based fuzzy inference system (ANFIS) approach for modelling hydrological time series. *Hydrological Sciences Journal*, 51(2), 588-598.
- Khan, M.S., and Coulibaly, P. (2006), Bayesian neural network for rainfall-runoff modeling. *Water Resources Research*, Vol. 42, W07409.
- Khu ST, Liong S-Y, Babovic V, Madsen H, Muttill N. (2001), Genetic programming and its application in real-time runoff forecasting. *Journal of the American Water Resources Association*, 37, 439-451.
- Khu, S.T., and Werner, M.G.F. (2003), Reduction of Monte-Carlo simulation runs for uncertainty estimation in hydrological modeling. *Hydrology and Earth System Sciences*, 7 (5), 680-692.
- Kim, T., Heo, J. H. and Jeong, C. S. (2006), Multireservoir system optimization in the Han River basin using multi-objective genetic algorithms. *Hydrological Processes*, 20 (9), 2057-2075.
- Kişi, Ö. (2003), River flow modeling using artificial neural networks. *Journal of Hydrologic Engineering*, 9(1), 60-63.
- Kişi, Ö. (2004), Multi-layer perceptrons with Levenberg-Marquardt training algorithm for suspended sediment concentration prediction and estimation. *Hydrological Sciences–Journal–des Sciences Hydrologiques*, 49(6), 1025-1040.
- Kişi, Ö. (2005), Daily river flow forecasting using artificial neural networks and auto-regressive models. *Turkish Journal of Engineering and Environmental Sciences*, 29, 9-20.
- Kitanidis, P. K. and Bras, R. L. (1980), Real-time forecasting with a conceptual hydrologic model, 2, applications and results, *Water Resources Research*, 16 (6), 1034–1044.
- Kojiri, T. and Ikeuchi, S. (1988), Real-time operation of dam reservoir by using fuzzy inference theory. The 6th Congress Asian and Pacific Regional Division International Association for Hydraulic Research, Kyoto, Japan, 1988: 437-443.
- Kojiri, T., Takasao, T., and Kanbayashi, Y. (1992), Fuzzy Expert System of Reservoir Operation with Typhoon and Rainfall Prediction[C].17th Int.Conf.on Applications of AI in Eng., Canada, 1992: 531-548.
- Kothyari, U.C. and Singh, V.P. (1999), Multiple input single output model for flow forecasting, *Journal of Hydrology*, 220, 12-26.
- Koutsoyiannis, D., and Pachakis, D. (1996), Deterministic chaos versus stochasticity in analysis and modeling of point rainfall series. *Journal of Geophysics Research*, 101 (D21), 26441-26451.
- Koza, J. R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA.

Krzysztofowicz, R. (2000), The case for probabilistic forecasting in hydrology. *Journal of Hydrology*, 249, 2-9.

Küçük, M. and Ağaloğlu, N. (2006), Wavelet Regression Technique for Stream flow Prediction. *Journal of Applied Statistics*, 33(9), 943-960.

Kuczera, G., and Parent, E. (1998), Monte Carlo assessment of parameter uncertainty in conceptual catchment models: The metropolis algorithm. *Journal of Hydrology*, 211, 69-85.

Kumar, A.R.S., Sudheer, K.P., Jain, S.K., and Agarwal, P.K. (2005), Rainfall-runoff modelling using artificial neural networks: comparison of network types. *Hydrological Processes*, 19 (6), 1277-1291.

Laio, F., Porporato, A., Revelli, R., and Ridolfi, L. (2003), A comparison of nonlinear flood forecasting methods, *Water Resources Research*, 39(5), 1129, doi:10.1029/2002WR001551.

Lampinen, J., and Vehtari, A. (2001), Bayesian approach for neural networks—review and case studies. *Neural Networks*, 14, 257– 274.

Laucelli, D., Giustolisi, O., Babovic, V. and Keijzer, M. (2007), Ensemble modeling approach for rainfall/groundwater balancing. *Journal of Hydroinformatics*, 9 (2), 95–106.

Lee, C.F., Lee, J.C. and Lee, A.C., (2000), Statistics for business and financial economics (2nd version), World Scientific, Singapore.

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

Li, Q.G., and Chen, S.Y. (2005), A SVM regress forecasting method based on the fuzzy recognition theory. *Advances in Water Science (in Chinese)*, 16(5), 741-746.

Li, Y., and Gu, R.R. (2003), Modeling Flow and Sediment Transport in a River System Using an Artificial Neural Network. *Environmental Management*, 31(1), 122-134.

Li, F., and Zeng, Q.C. (2008), Statistical prediction of East Asian summer monsoon rainfall based on SST and sea ice concentration. *Journal of the meteorological society of Japan*, 86 (1), 237-243.

Liang, G.C., Kachroo, R.K., Kang, W., and Yu, X.Z.(1992), River flow forecasting. Part 4. Applications of linear modeling techniques for flow routing on large catchments. *Journal of Hydrology*, 133, 99-140.

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

Lin, J.Y., Cheng, C.T. and Chau, K.W. (2006), Using support vector machines for long-term discharge prediction. *Hydrological Sciences Journal*, 51(4), 599-611.

Lindskog, P. (1997), Fuzzy identification from a grey box modeling point of view. In Fuzzy Model Identification: Selected Approaches (ed. H. Hellendoorn & D. Driankov), pp. 3–50. Springer-Verlag, Berlin.

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

Liong, S. Y., Gautam, T. R., Khu, S. T., Babovic, V., and Muttal, N. (2002), Genetic programming: A new paradigm in rainfall-runoff modeling. *Journal of American Water Resources Association*, 38(3), 705-718.

Liong, S.Y., Lim, W.H., Kojiri, T., and Hori, T. (2000), Advance flood forecasting for flood stricken Bangladesh with a fuzzy reasoning method. *Hydrological Processes*, 14, 431-448.

Liong, S.Y., and Sivapragasam, C. (2002), Flood stage forecasting with support vector machines. *Journal of American Water Resources*, 38(1), 173 -186.

Lisi, F., Nicolis, and Sandri, M. (1995), Combining singular-spectrum analysis and neural networks for time series forecasting. *Neural Processing Letters*, 2(4), 6-10.

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

Luk, K.C., Ball, J.E., and Sharma, A. (2000), A study of optimal model lag and spatial inputs to artificial neural network for rainfall forecasting. *Journal of Hydrology*, 227, 56-65.

Mack, Y. P. and Rosenblatt, M. (1979), Multivariate k-nearest neighbor density estimates. *Journal of Multivariate Analysis*, 9, 1-15.

Mahabir, C., Hicks, F.E, and Fayek, A. R. (2003), Application of fuzzy logic to forecast seasonal runoff . *Hydrological Processes*, 17, 3749-3762.

Maier, H.R., and Dandy, G.C. (1997), Determining inputs for neural network models of multivariate time series. *Microcomputers in Civil Engineering* 12 (5), 353-368.

Maier H.R., and Dandy G.C. (2000), Neural networks for the prediction and forecasting of water resources variables: a review of modeling issues and applications. *Environmental Modeling and Software*, 15, 101-23.

Maier, H.R., Dandy, G.C., and Burch, M.D. (1998), Use of artificial neural networks for modelling cyanobacteria Anabaena spp. in the River Murray, South Australia. *Ecological Modelling*, 105 (2/3), 257-272.

Mallat, S.G., (1989), A theory for multi-resolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7), 674-692.

Mamdani, E. H. (1974), Application of fuzzy algorithms for control of simple dynamic plant. *Proc. IEE*, 121 (12), 1585-1588.

Marques, C.A.F., Ferreira, J., Rocha, A., Castanheira, J., Gonçalves, P., Vaz. N., and Dias, J.M. (2006), Singular spectral analysis and forecasting of hydrological time series. *Physics and Chemistry of the Earth*, 31, 1172-1179.

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

Master, T. (1993), *Practical Neural Networks Recipes in C + +*. Academic Press, San Diego, CA.

May, R.J., Maier, H.R., Dandy, G.C., and Fernando, T.M.K. (2008), Non-linear variable selection for artificial neural networks using partial mutual information. *Environmental Modeling & Software*, 23, 1312-1328.

Minns, A. W. (1998), *Artificial Neural Networks as Subsymbolic Process Descriptors*. Balkema, Rotterdam, The Netherlands.

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

Mitchell, T.M. (1997), *Machine learning*. McGraw-Hill, New York.

Montanari, A., and Brath, A. (2004), A stochastic approach for assessing the uncertainty of rainfall-runoff simulations, *Water Resources Research*, 40, W01106, doi:10.1029/2003WR002540.

Moncada, E., Fontane, D.G., and Gates, T.K. (1994), Evaluating multi-purpose reservoir operations using fuzzy dynamic programming, pp111-114. Proceedings of the 21st Annual Conference sponsored by the Water Resources Planning and Management Division, ASCE, May 23-26, Denver, Colorado.

Mulvany, T. J. (1850), On the use of self-registering rain and flood gauges. *Proc. Inst. Civ. Eng.*, 4(2), 1-8.

Munot, A. A., and Kumar, K.K. (2007), Long range prediction of Indian summer monsoon rainfall. *Journal of Earth System Science*, 116 (1), 73-79.

Muster, H., Bárdossy, A., and Duckstein, L. (1994), Adaptive neuro-fuzzy modeling of a non-stationary hydrologic variable.

Mutlu, E., Chaubey, I., Hexmoor, H., and 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.

Mutti, N. and Chau. K.W. (2006), Neural network and genetic programming for modelling coastal algal blooms. *International Journal of Environment and Pollution*, 28 (3/4), 223-238.

Nash, J. E., Barsi, B. I., (1983), A Hybrid Model for Flow Forecasting on Large Catchments. *Journal of Hydrology*, 65(1-3), 125-137.

- Nash, J. E. and Sutcliffe, J. V. (1970), River flow forecasting through conceptual models part I — A discussion of principles. *Journal of Hydrology*, 10 (3), 282-290.
- Natale L. and Todini E., (1976), Black box identification of a linear flood wave propagation model, in Singh P. V., (ed.), (1982), "Rainfall-Runoff relationship", Water Resources Publications, USA.
- Nayagam, L.R., Janardanan, R., and Mohan, H.S.R. (2008), An empirical model for the seasonal prediction of southwest monsoon rainfall over Kerala, a meteorological subdivision of India. *International Journal of Climatology*, 28 (6), 823-831.
- Nayak, P.C., Sudheer, K.P., and Jain, S.K. (2007), Rainfall-runoff modeling through hybrid intelligent system. *Water Resources Research*, 43 (7), W07415, doi:10.1029/2006WR004930.
- Nayak, P. C., Sudheer, K. P., and Ramasastri, K. S. (2004), Fuzzy computing based rainfall-runoff model for real time flood forecasting. *Hydrological Processes*, 19, 955-968.
- Nayak, P.C., Sudheer,K.P., Rangan, D.M. and Ramasastri, K.S. (2005), Short-term flood forecasting with a neuro-fuzzy model, *Water Resources Research*, 41, 1-16 (W04004).
- Newbold, P., Carlson, W. L., and Thorne, B.M. (2003), Statistics for business and economics (fifth version), Prentice Hall, Upper Saddle River, N.J.
- Nord, L.I., and Jacobsson, S.P. (1998), A novel method for examination of the variable contribution to computational neural network models. *Chemometrics and Intelligent Laboratory Systems*, 44(1), 153-160.
- Novák ,V., Perfilieva, I., and Mockor, J. (1999), *Mathematical Principles of Fuzzy Logic*. Boston: Kluwer Academic Publishers.
- O'Connor, K. M. (2005), River flow forecasting. In *River Basin Modelling for Flood Risk Mitigation* (ed. D. W. Knight & A. Y. Shamseldin), pp. 197-213. Taylor & Francis, London.
- Papadokonstantakis, S., Lygeros, A., and Jacobsson, S. P. (2006), Comparison of recent methods for inference of variable influence in neural networks. *Neural Networks*, 19(4), 500-513.
- Partal T. (2009), River flow forecasting using different artificial neural network algorithms and wavelet transform. *Canadian Journal of Civil Engineering*, 36(1), 26-39.
- Partal. T. and Cigizoglu, H.K. (2008), Estimation and forecasting of daily suspended sediment data using wavelet-neural networks, *Journal of Hydrology*, 358, (3-4), 317-331.
- Partal, T. and Kişi, Ö. (2007), Wavelet and Neuro-fuzzy conjunction model for precipitation forecasting. *Journal of Hydrology*, 342 (2), 199-212.

- Pasternack, G.B. (1999). Does the river run wild? Assessing chaos in hydrological systems. *Advance in Water Resources*, 23, 253–260.
- Penman, H. L. (1961), Weather, plant and soil factors in hydrology. *Weather*, 16, 207–219.
- Pearson, K., (1901), On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2, 559-572.
- Phoon, K. K., Islam, M. N., Liaw, C. Y. and Liang, S. Y. (2002), Practical inverse approach for forecasting nonlinear hydrological time series. *Journal of Hydrological Engineering ASCE*, 7 (2), 116-128.
- Pongracz, R., Bartholy, J., and Bogardi, I. (2001), Fuzzy rule-based prediction of monthly precipitation. *Physics and Chemistry of the Earth Part B-Hydrology Oceans and Atmosphere*, 26 (9), 663-667.
- Porporato, A., and Ridolfi, L. (1997), Nonlinear analysis of river flow time sequences. *Water Resources Research*, 33 (6), 1353-1367.
- Procaccia, I. (1988), Complex or just complicated? *Nature*, 333, 498-499.
- Prochazka, A. (1997), Neural networks and seasonal time-series prediction. *Artificial Neural Networks, Fifth International Conference on* (Conf. Publ. No. 440), Vol., Iss., 36-41.
- Pulido-Calvo, I. and Portela, M.M. (2007), Application of neural approaches to one-step daily flow forecasting in Portuguese watersheds. *Journal of Hydrology*, 332, 1-15.
- Qin, G.H., Ding, J., and Liu, G.D. (2002) River flow prediction using artificial neural networks: self-adaptive error back-propagation algorithm. *Advances in Water Science (in Chinese)*, 13 (1), 37-41.
- Rabuñal, J.R., Dorado, J., Puertas, J., Pazos, A., Santos, A., and Rivero, D. (2003), Prediction and modeling of the rainfall-runoff transformation of a typical urban basin using ANN and GP. *Applied Artificial Intelligence*, 17(4), 329-343.
- Rajurkar, M.P., Kothiyari, U.C., and Chaube, U.C. (2002), Artificial neural networks for daily rainfall-runoff modeling. *Hydrological Sciences-Journal*, 47(6), 865-876.
- Raman, H., and Chandramouli, V. (1996), Deriving a general operating policy for reservoirs using neural networks. *ASCE Journal of Water Resources Planning and Management*, 122 (5), 342-347.
- Raman, H., and Sunilkumar, N. (1995), Multivariate modeling of water resources time series using artificial neural networks. *Hydrological Sciences Journal*, 40(2):145-163.
- Ripley, B.D. (1994), Neural networks and related methods of classification. *Journal of the Royal Statistical Society B*, 56 (3), 409-456.

- Roadknight, C.M., Balls, G.R., Mills, G.E., and Palmer-Brown, D. (1997), Modeling complex environmental data. *IEEE transactions on neural networks*, 8(4), 852-862.
- Robinson Fayek, A., and, Sun Z. (2001), A fuzzy expert system for design performance prediction and evaluation. *Canadian Journal of Civil Engineeringm*, 28, 1-25.
- Rodriguez-Iturbe, I., Febres De Power, B., Sharifi, M. B., and Georgakakos, K. P. (1989), Chaos in Rainfall, *Water Resources Research*, 25(7), 1667-1675.
- Ruelle, D. (1990), *Proc. R. Soc. London, Ser. A* 427, 241.
- Rumelhart, D. E., McClelland, J. L., and the PDP research group. (1986), *Parallel distributed processing: Explorations in the microstructure of cognition. Volume I*. Cambridge, MA: MIT Press.
- Sahai, A.K., Soman, M.K., and Satyan, V. (2000), All India summer monsoon rainfall prediction using an artificial neural network. *Climate Dynamics*, 16 (4), 291-302.
- Sajikumar, N., and Thandaveswara, B.S. (1999), A non-linear rainfall-runoff model using artificial neural networks. *Journal of Hydrology*, 216, 32-55.
- Salas, J.D., Delleur, J.W., Yevjevich, V., and Lane, W.L. (eds) (1985), *Applied Modeling of Hydrologic Time Series*. Water Resources Publications: Littleton, Colorado.
- Salas, J.D., Markus, M., and Tokar, A.S. (2000), Stream flow forecasting based on artificial neural networks. In *artificial neural networks in hydrology*, (eds.) Govindaraju, R.S., and Rao, A.R. Water Science and Technology Library.
- Sarle, W.S. (1995), Stopped training and other remedies for overfitting, paper presented at the 27th symposium on the interface for computing science and statistics, Carnegie Mellon University, Pittsburgh, Pa.
- Sarle, W.S., (1994), Neural networks and statistical models. In: *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, pp. 1538–1550. SAS Institute.
- Sauer, T., Yorke, J. A., and Casdagli, M., (1991), Embedology. *Journal of Statistics and Physics*, 65, 579- 616.
- Savic, D.A., Walters, G.A., and Davidson, J.W., (1999), A genetic programming approach to rainfall-runoff modeling. *Water Resources and Management*, 13, 219-231.
- Schalkoff, R. J., (1997), *Artificial Neural Networks*, McGraw-Hill Higher Education.
- Schertzer, Tchiguirinskaia, D., I., Lovejoy, S., Hubert, P., Bendjoudi, H., and Larchevêque, M. (2002), Which chaos in the rainfall runoff process? A discussion on Evidence of chaos in the rainfall-runoff process by Sivakumar. *Hydrological Sciences Journal*, 47 (1), 139-147.

Soofi, E.S., Retzer, J.J., (2003), Information importance of explanatory variables. In: IEE Conference in Honor of Arnold Zellner: Recent Developments in the Theory, Method and Application of Entropy Econometrics, Washington.

Scott, D.W. (1992), Multivariate Density Estimation: Theory, Practice and Visualisation. John Wiley and Sons, New York.

Sebzalli Y.M., and Wan, X.Z. (2001), Knowledge discovery from process operational data using PCA and fuzzy clustering. Engineering Applications of Artificial Intelligence ,14, 607-616.

See L, and Abrahart, R J. (2001), Multi-model data fusion for hydrological forecasting. Computers and Geosciences, 27(10), 987-994.

See L, and Openshaw, S. (1999), Applying soft computing approaches to river level forecasting. Hydrology Science Journal, 44(5), 763-778.

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

Seno, E., Izumida, M., Murakami, K., and Matsumoto, S. (2003), Inflow forecasting of the dam by the neural network. ANNIE, 701-706.

Shamseldin, A.Y. (1997), Application of a neural network technique to rainfall-runoff modelling. Journal of Hydrology 199, 272-294.

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

Shamseldin, A.Y. and O'Connor, K.M. (1999), A real-time combination method for the outputs of different rainfall-runoff models. Hydrological Sciences Journal, 44(6), 895-912.

Shamseldin, A.Y., O'Connor, K.M. and Liang, G.C., (1997), Methods for combining the outputs of different rainfall runoff models, Journal of Hydrology, 197, 203–229.

Shamseldin, A.Y., O'Connor, K.M., and Nasr, A.E. (2007), A comparative study of three neural network forecast combination methods for simulated river flows of different rainfall-runoff models. Hydrological Sciences Journal-Journal des Sciences Hydrologiques, 52 (5), 896-916.

Shao, R., Zhang J., Martin, E. B., and Morris, A. J. (1997), Novel approaches to confidence bound generation for neural network representation. Artifical Neural Networks, 7-9 July,1997, Conference Publication No. 440.

Shao, R., Zhang J., Martin, E. B., and Morris, A. J. (1997), Novel approaches to confidence bound generation for neural network representation. Artifical Neural Networks, 7-9 July,1997, Conference Publication No. 440.

Sharma, A. (2000), Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 - a strategy for system predictor identification. Journal of Hydrology 239, 232-239.

- Sheng, C., Gao, S., and Xue, M. (2006), Short-range prediction of a heavy precipitation event by assimilating Chinese CINRAD-SA radar reflectivity data using complex cloud analysis. *Meteorology and Atmospheric Physics*, 94 (1-4), 167-183.
- Shrestha, B. P., Duckstein, L. E. and Stokhin, Z., (1997), Fuzzy rule-based modeling of reservoir Operation. *Journal of Water Resources Planning and Management*, 122(4), 262-269.
- Shrestha, D. L., N. Kayastha, and D. P. Solomatine, (2009), A novel approach to Parameter uncertainty analysis. *Hydrology and Earth System Sciences*, 6, 1677-1706.
- Shrestha, D.L., and Solomatine, D.P. (2006), Machine learning approaches for estimation of prediction interval for the model output. *Neural Networks*, 19(2), 225-235.
- Shu, C. and Burn, D.H. (2004), Homogeneous pooling group delineation for flood frequency analysis using a fuzzy expert system with genetic enhancement. *Journal of Hydrology*, 291, 132-149.
- Sheta, A.F., and El-Sherif, M.S. (1999), Optimal prediction of the Nile River flow using neural networks. *Neural Networks*, 1999. IJCNN '99. International Joint Conference on, 5, 3438-3441.
- Silverman, D., and Dracup, J.A. (2000), Artificial neural networks and long-range precipitation in California. *Journal of Applied Meteorology* 31 (1), 57-66.
- Singh, V.P. and Woolhiser, D.A. (2002), Mathematical modeling of watershed hydrology. *Journal of Hydrologic Engineering*, 7(4), 270-292.
- Sivakumar, B., Berndtsson, R., and Persson, M. (2001), Monthly runoff prediction using phase-space reconstruction. *Hydrological Science Journal*, 46 (3), 377-388.
- Sivakumar, B., Jayawardena, A. W., and Fernando, T. M. K. (2002), River flow forecasting: use of phase-space reconstruction and artificial neural networks approaches. *Journal of Hydrology*, 265(1), 225-245.
- Sivakumar, B., Liong, S.Y. and Liaw, C.Y. (1998), Evidence of chaotic behavior in Singapore rainfall. *Journal of the American water resources association*, 34(2), 301-310.
- Sivapragasam, C., Liong, S.Y. and Pasha, M.F.K. (2001), Rainfall and runoff forecasting with SSA-SVM approach. *Journal of Hydroinformatics*, 3(7), 141-152.
- Sivapragasam, C., and Liong, S. Y. (2005), Flow categorization model for improving forecasting. *Nordic Hydrology* 36 (1), 37-48.
- Sivapragasam, C., Vincent and, P., and Vasudevan, G. (2007), Genetic programming model for forecast of short and noisy Data. *Hydrological Processes*, 21, 266-272.
- Smaoui, N, and Al-Enezi, S.(2004), Modelling the dynamics of nonlinear partial differential equations using neural networks. *Journal of Computational and Applied Mathematics*, 170, 27-58.

Smith, J., and Eli, R.N. (1995), Neural-network models of rainfall-runoff process. Journal of Water Resources Planning and Management, 121(6), 499-508.

Solomatine, D.P., Rojas, C., Velickov, S. and Wust, H. (2000), Chaos theory in predicting surge water levels in the North Sea. In: Proc. 4th Int. Conf. on Hydroinformatics, Cedar Rapids.

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

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

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

Solomatine, D.P., and Shrestha, D.L. (2009), A novel method to estimate model uncertainty using machine learning techniques. Water Resources Research, 45, W00B11, doi:10.1029/2008WR006839.

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

Sudheer, K. P. (2004), Knowledge extraction from trained neural network river flow models. Journal of Hydrologic Engineering, ASCE, 10(4), 264-269.

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

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

Sudheer, K.P. and Jain, S.K. (2003), Radial basis function neural network for modeling rating curves. ASCE. Journal of Hydrologic Engineering, 8(3), 161-164.

Sugeno M., and Kang, G. T. (1988), Structure identification of fuzzy model. Fuzzy Sets and Systems, 28(1), 15-33.

Sugihara, G and May, R.M. (1990), Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. Nature, 344, 734- 741.

Takagi, T. and Sugeno, M. (1985), Fuzzy identification of systems and its application to modeling and control. IEEE Trans. Syst. Man Cybern. SMC-15 (1), 116-132.

Takens, F., (1981), Detecting strange attractors in turbulence. In: Rand, D.A., Young, L.S. (Eds.), Dynamical Systems and Turbulence, Lecture Notes in Mathematics 898, Springer, Berlin, pp. 366-381.

- Tanaka, K. (1996), An Introduction to Fuzzy Logic for Practical Applications (translated by Niimura, T.). Springer.
- Tayfur, G., and Singh, V.P. (2006), ANN and Fuzzy Logic Models for Simulating Event-Based Rainfall-Runoff. *Journal of Hydraulic Engineering*, 132(12), 1321-1330.
- Theiler, J., (1986), Spurious dimension from correlation algorithms applied to limited time-series data. *Physical Review A*, 34 (3), 2427-2432.
- Thirumalaiah, K., and Deo, M.C. (1998), River stage forecasting using artificial neural networks. *Journal of Hydrologic Engineering*, 3(1), 26-32.
- Thirumalaiah, K., and Deo, M.C. (2000), Hydrological forecasting using neural networks. *Journal of Hydrologic Engineering*, 5(2), 180-189.
- Thrift, P. (1991), Fuzzy Logic Synthesis with Genetic Algorithms. In Proc. 4th Int. Conf. Genetic Algorithms (ICGA), 509-513, San Diego, CA.
- Tibshirani, R. (1996), A comparison of some error estimates for neural network models. *Neural Computation*, 8(1), 152-153.
- Todini E. (1996), The arno rainfall-runoff model. *Journal of Hydrology*, 175, 339-382.
- Torrence, C. and Compo, G.P. (1998), A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, 79, 61-78.
- Toth, E., and Brath, A. (2007), Multistep ahead streamflow forecasting: Role of calibration data in conceptual and neural network modeling. *Water Resources Research*, 43(11), art. no. W11405.
- Tokar, A.S., and Johnson, P.A. (1999), Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, 4(3), 232-239.
- Tokar, A.S., and Markus, M. (2000), Precipitation-runoff modeling using artificial neural networks and conceptual models. *Journal of Hydrologic Engineering*, 5, 156-161.
- Toth, E, Brath, A., and Montanari, A. (2000), Comparison of short-term rainfall prediction models for real-time flood forecasting. *Journal of Hydrology*, 239, 132-147.
- Tsonis, A.A. (1992), Chaos: From Theory to Applications, Plenum Press, New York.
- Valdés, M., Gómez-Skarmeta, A.F., and Botía, J.A. (2005), Toward a Framework for the Specification of Hybrid Fuzzy Modeling. *International Journal of Intelligent Systems*, 20(2), 225-252.
- Van Veldhuizen, D. A. and Lamont, G. B. (2000), Multiobjective evolutionary algorithms analyzing the state-of-the-art. *Journal of Evolutionary Computation*, 8(2), 125–144.

- Vapnik, V. (1995), *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.
- Vapnik, V. N. (1998), *Statistical Learning Theory*. John Wiley & Sons, New York.
- Vapnik, V.N. (1999), An overview of statistical learning theory. *IEEE Transactions on Neural Networks* 10 (5), 988-999.
- Varoonchotikul, P. (2003), Flood forecasting using artificial neural networks, Ph.D. thesis, Swets & Zeitlinger, Lisse, The Netherlands.
- Vautard, R., and Ghil, M. (1989), Singular-spectrum analysis in nonlinear dynamics, with applications to paleoclimatic time series. *Physica D*, 35, 395-424.
- Vautard, R., Yiou, P. and Ghil, M. (1992), Singular-spectrum analysis: a toolkit for short, noisy and chaotic signals. *Physica D* 58, 95-126.
- Venkatesan, C., Raskar, S.D., Tambe, S.S., Kulkarni, B.D., and Keshavamurty, R.N. (1997), Prediction of all India summer monsoon rainfall using error-back-propagation neural networks. *Meteorology and Atmospheric Physics*, 62 (3-4), 225-240.
- Vernieuwe, H., Georgieva O., De Baets, B., Pauwels, V.R.N., Verhoest N.E.C., and De Troch F.P. (2005), Comparison of data-driven Takagi-Sugeno models of rainfall-discharge dynamics, *Journal of Hydrology*, 302, 173-186.
- Vojinovic, Z., Kecman, V. and Sharman, B. (2002), A radial basis function neural network model as an error correction tool for ICS deterministic models. In: *Hydroinformatics 2002* (Proc. Fifth Int. Conf. on Hydroinformatics) (Cardiff, UK), 659–665. IWA Publishing, London, UK.
- Wagener, T., Wheatear, H. S., and Gupta, H. V. (eds.) (2004), *Rainfall-runoff modeling in gauged and ungauged catchments*. London: Imperial College Press.
- Wang, W., Van Gelder, P.H.A.J.M., and Vrijling, J.K., (2005), Some issues about the generalization of neural networks for time series prediction. In: Duch, W. (Ed.), *Artificial Neural Networks: Formal Models and Their Applications*, Lecture Notes in Computer Science, 3697, 559–564.
- Wang,W., Van Gelder, P.H.A.J.M., Vrijling, J.K. and Ma, J. (2006a), Testing for nonlinearity of streamflow processes at different timescales. *Journal of Hydrology*, 322, 247-268.
- Wang,W., Van Gelder, P.H.A.J.M., Vrijling, J.K. and Ma, J. (2006b), Forecasting Daily Streamflow Using Hybrid ANN Models. *Journal of Hydrology*, 324, 383-399.
- Wilby, R.L., Abrahart, R.J., and Dawson, C. W. (2003), Detection of conceptual model rainfall-runoff processes inside an artificial neural network. *Hydrological Sciences Journal*, 48(2), 163-181.
- Wolf, A. Swift, J. B., Swinney, H. L., and Vastano, J. A. (1985), Determining Lyapunov exponents from a time series. *Physica D* 16, 285-317.

- Wolpert, D.H. (1992), Stacked generalization. *Neural networks*, 5, 241-259.
- Wu, C.L., Chau, K.W. and Li, Y.S. (2008), River stage prediction based on a distributed support vector regression. *Journal of Hydrology*, 358, 96-111.
- Wu, J. S., Han, J., Annambhotla, S., and Bryant, S. (2005), Artificial Neural Networks for Forecasting Watershed Runoff and Stream Flows. *Journal of Hydrologic Engineering*, 10(3), 216-222.
- Xiong, L.H., Shamseldin, A.Y., and O'Connor, K.M. (2001), A non-linear combination of forecasts of rainfall-runoff models by the first-order TS fuzzy system for forecast of rainfall-runoff model. *Journal of Hydrology*, 245, 196-217.
- Xu, Z.X., and Li, J.Y. (2002), Short-term inflow forecasting using an artificial neural network model. *Hydrological Processes*, 16(12), 2423-2439.
- Yakowitz, S. (1987), Nearest neighbor method for time series analysis. *Journal of Time Series Analysis*, 8 (2), 235-247.
- Yang, D., Herath, S., and Musiak, K. (1998), Development of a geomorphology-based hydrological model for large catchments. *Annual Journal of Hydraulic Engineering*, 42, 169-174.
- Yates, D.N., Warner, T.T., and Leavesley, G.H. (2000), Prediction of a flash flood in complex terrain. Part II: A comparison of flood discharge simulations using rainfall input from radar, a dynamic model, and an automated algorithmic system. *Journal of Applied Meteorology*, 39 (6), 815-825.
- Yu, P.S. and Chen, S.T. (2005), Updating real-time flood forecasting using a fuzzy rule-based model. *Hydrological Sciences Journal*, 50(2), 265-278.
- Yu, P.S., Chen, S.T., and Chang I.F. (2006), Support vector regression for real-time flood stage forecasting. *Journal of hydrology*, 328, 704-716.
- Yu, D.L., Gomm, J.B., and Williams, D. (2000), Neural model input selection for a MIMO chemical process. *Engineering Applications of Artificial Intelligence*, 13(1), 15-23.
- Yu, X.Y., Liong, S.Y., and Babovic, V. (2004), EC-SVM approach for real-time hydrologic forecasting. *Journal of Hydroinformatics*, 6(3), 209-233.
- Yu, D.J., Small, M., Harrison, R.G. and Diks, C. (2000), Efficient implementation of the Gaussian kernel algorithm in estimating invariants and noise level from noisy time series data. *Physical Review E*, 61, 3750-3756.
- Yu, P.S. and Tseng, T.Y. (1996), A model to forecast flow with uncertainty analysis. *Hydrological Sciences Journal*, 41(3), 327-344.
- Zealand, C.M. Burn, D.H. and Simonovic, S.P. (1999), Short term streamflow forecasting using artificial neural networks. *Journal of Hydrology*, 214, 32-48.
- Zhang, M. (1999). Genetic algorithm based neural network model for forecasting flood. *Journal of Dalian Institute of Light industry (in Chinese)*, 18(2), 168-173.

Zhang, B., and Govindaraju, R. S. (2000), Prediction of watershed runoff using Bayesian concepts and modular neural networks. *Water Resources Research*, 36(3), 753-762.

Zhao, R.J. (1992), The Xinanjiang model applied in China. *Journal of Hydrology*, 135(1-4), 371-381.

Zou, R., Lung, W.S., and Guo, H.C. (2002), Neural Network Embedded Monte Carlo Approach for Water Quality Modeling under Input Information Uncertainty. *Journal of Computing in Civil Engineering*, 16(2), 135-142.

Zadeh, L. A. (1965), Fuzzy sets. *Inf. Control*, 8 (3), 338-353.

Zadeh, L. A. (1994), Soft computing and fuzzy logic. *IEEE Transactions on Software*, 11(6), 48-56.

Zealand, C. M., Burn, D. H., and Simonovic, S. P. (1999), Short term stream flow forecasting using artificial neural networks. *Journal of Hydrology*, 214, 32-48.

Zheng, G.L., and Billings, S.A. (1996), Radial Basis Function Network Configuration Using Mutual Information and the Orthogonal Least Squares Algorithm. *Neural Networks*, 9(9), 1619-1637.