HYBRID MODEL OF NEURAL NETWORK AND POPULATION-BASED OPTIMIZATION ALGORITHM FOR RIVER FLOW AND SEDIMENT LOAD

XIAOYUN CHEN

Ph.D

The Hong Kong Polytechnic University

2018

The Hong Kong Polytechnic University

Department of Civil and Environmental Engineering

Hybrid Model of Neural Network and Population-based Optimization Algorithm for River Flow and Sediment Load

Xiaoyun Chen

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

October 2017

Certificate of Originality

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

Signature: _____

Name: Xiaoyun Chen

Abstract

River flow forecasting and sediment load estimation are important issues in the hydrological field, and customarily undertaken by data-driven models. Despite the amount of research on the subject, most of them are incapable of providing insight into the unrecognized relationship of the input and output variables owing to a black-box nature. It is inevitable that the physically-meaningless models suffer from inappropriate input-output mapping, substantial uncertainty inherent in the modeling and inadequate optimization during calibration. This thesis is an attempt to develop physics-based models for efficiently and reliably simulating the river flow and sediment load.

A novel hybrid neural network (HNN) model is proposed for downstream river flow forecasting, by combining fuzzy pattern-recognition and continuity equation in a neural network. The model is therefore, able to reflect fuzzy and time-varying features of river flows. In comparison with three benchmarking models, the HNN model is identified as the preferred tool to fit the total observations. The superiority of HNN model does not markedly deteriorate with the increase of forecasting lead time.

With respect to sediment load estimation, a hybrid double feedforward neural network (HDFNN) model is developed by integrating fuzzy pattern-recognition and continuity equation into a structure of double neural networks. The generalization and estimation abilities of HDFNN models are verified by comparison with results from its counterpart models. It could reproduce medium and high loads appropriately, and present excellent performances in multi-step-ahead estimations particularly for high loads.

The request of determining the best input variables for the HNN and HDFNN models has been processed. Generally, areal precipitation is an appropriate input variable coupled with all observed upstream flows for the HNN model. The input variables may be uncertain and unstable when forecast lead time is shorter than the flow travel time. For the sediment estimation, river flows from the upstream and downstream stations with different ahead of times are selected to formulate input combinations. It is found that the downstream sediment seems to be more sensitive to upstream flows with small studied area while the downstream flows substantially affects the high sediment loads.

The lower upper bound estimation (LUBE) is a straight-forward method that could construct the neural network based models with two output neurons and directly approximate the lower and upper bounds of prediction intervals. Applications on the HNN and HDFNN models indicate their reliability in hydrological prediction scenarios.

The performances of three population-based optimization algorithms, namely differential evolution (DE), artificial bee colony (ABC) and ant colony optimization (ACO) for evolving the HNN and HDFNN models are compared. The DE is found to be a more appropriate algorithm in terms of generalization and prediction. The ABC appears to be more adapted in optimizing the multi-step-ahead cases, but on the other hand, presents computational inefficiency. As far as the stability is concerned, the DE and ABC algorithms are more adaptive than the ACO with the population size.

The major contribution of this research is the development of HNN and HDFNN models for river flow and sediment load. The LUBE method has proven as a promising technique to evaluate the model reliability. In addition, this thesis advocates the use of DE algorithm for the optimization problems in hydrological models.

Publications

Articles in Journals

Chen, X. Y. and Chau, K. W. (2016). A hybrid double feedforward neural network for suspended sediment load estimation. Water Resources Management, 30(7), 2179-2194.

Chen, X. Y., Chau, K. W. and Busari, A. O. (2015). A comparative study of population-based optimization algorithms for downstream river flow forecasting by a hybrid neural network model. Engineering Applications of Artificial Intelligence, 46, 258-268.

Chen, X. Y., Chau, K. W. and Wang, W. C. (2015). A novel hybrid neural network based on continuity equation and fuzzy pattern-recognition for downstream daily river discharge forecasting. Journal of Hydroinformatics, 17(5), 733-744.

Chen, X. Y. and Chau, K. W. (2017). Uncertainty prediction for suspended sediment load using hybrid neural network model and LUBE method. Water Resources Management. (submitted for publication)

Conference posters

Chen, X. Y. and Chau, K. W. (2015). Downstream river flow forecasting by hybrid neural network model with upstream flow and precipitation inputs. The 36th IAHR World Congress. 28 June-3 July, 2015. Delft-The Hague, The Netherlands.

Acknowledgments

First of all, I would like to express my sincere appreciation to my supervisor, Prof. K.W. Chau, for his constant encouragement, guidance and support. It would be impossible to complete the research without his patient instructions and insightful suggestions.

The financial supports by the Central Research Grant of Hong Kong Polytechnic University (4-ZZAD) and The Hong Kong Polytechnic University are greatly acknowledged. Without the support of these organizations, the research project is impossible to complete.

Secondly, I also wish to acknowledge the support from Dr. W.C. Wang and Dr. R. Taormina with whom I started this research topic. I would like to express my sincere appreciation to Prof. K.T. Chau, Prof. C.W. Li, Prof. W.H. Wai and Dr. Robina Wong for their valuable suggestions and help on the research program.

Then I would like to extend my thanks to all the Faculty members and researchers with whom I had the honor to work during my Ph.D studies. In particular, I wish to express my sincere appreciation to the Dr. A.O. Busari and Mr. X.F. Yan with whom I attended the 36th IAHR World Congress in the Netherlands. I also wish to thank my friends P.F. Cheng and L.Tang who encouraged me in times of hardship.

Finally, I wish to thank my dear parents and my husband for their love, support and encouragement on my study.

Table of Contents

List of Figures

List of Tables

Abbreviation

ABC	Artificial Bee Colony
AC	Auto Correlation
ACC	Accuracy
ACO	Ant Colony Optimization
ANFIS	Adaptive Neuro-Fuzzy Inference System
ANN	Artificial Neural Network
AP	Areal Precipitation
AR	Auto-Regressive
ARMA	Auto-Regressive Moving Average
BNN	Bayesian Neural Network
BP	Back Propagation
CC	Cross Correlation
CL	Confidence Level
CWC	Coverage Width-based Criterion
DE	Differential Evolution
DM	Diebold-Mariano
DPFNN	Double Parallel Feedforward Neural Network
EA	Evolutionary Algorithms
FFBP	FeedForward Back Propagation
FIPS	Fully Informed Particle Swarm
FIS	Fuzzy Inference Systems
FPNN	Fuzzy Pattern-recognition Neural Network
GA	Genetic Algorithm
GEP	Gene Expression Programming
GKF	Geno-Kalman Filtering
GLUE	Generalized Likelihood Uncertainty Estimation
GP	Genetic Programming
GRNN	Generalized Regression Neural Network
HDFNN	Hybrid Double Feedforward Neural Network
HEC-HMS	Hydrologic Engineering Center-Hydrologic Modeling System

HFNN	Hybrid Feedforward Neural Network
HNN	Hybrid Neural Network
HYMN	Hydrological Modeling Network
IHACRES	Identification of unit Hydrographs And Component flows from Rainfall, Evaporation and Streamflow
K-NN	K-Nearest-Neighbor
LGP	Linear Genetic Programming
L-M	Levenberg Marquardt
LSSVM	Least Square Support Vector Machine
LUEM	Local Uncertainty Estimation Model
LUBE	Lower Upper Bound Estimation
MFNN	Multi-layer Feedforward Neural Network
MI	Mutual Information
MLP	Multi-Layer Perceptron
MLR	Multi-Linear Regression
MNLR	Multiple Non-Linear Regression
NSEC	Nash-Sutcliffe Efficiency Coefficient
NSGA	Non-dominated Sorting Genetic Algorithm
NF	Neuro-Fuzzy
PAC	Partial Auto Correlation
PCA	Principal Component Analysis
PDM	Probability Distributed Model
PI	Prediction Interval
PICP	Prediction Interval Coverage Probability
PINAW	Prediction Interval Normalized Average Width
PINRW	Prediction Interval Normalized Root-mean-square Width
PMI	Partial Mutual Information
PP	Point Precipitation
PSACO	Particle Swarm Ant Colony Optimization
PSO	Particle Swarm Optimization
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
SAC-SMA	Sacramento Soil Moisture Accounting
SCE	Shuffled Complex Evolution

SRC	Sediment Rating Curve
SSA	Singular Spectrum Analysis
SSC	Suspended Sediment Concentration
SSL	Suspended Sediment Load
SVM	Support Vector Machine
TRMM	Tropical Rainfall Measuring Mission
UNEEC	Uncertainty Estimation based on local Error and Clustering
USGS	United States Geological Survey
WA	Wavelet Analysis
WANN	Wavelet Artificial Neural Network
WNF	Wavelet transform and Neuro-Fuzzy
WSVM	Wavelet Support Vector Machine
XAJ	Xinanjiang

1 Introduction

1.1 Research background

The assessments of flow and sediment in a river system are of vital interest in hydrology and hydraulic engineering, owing to their applications in the design and management of water resource projects. In order to control flow and sediment discharges more efficiently, models that forecast flow and sediment are desired to be of high precision. The problem is customarily settled with data-driven models, of which artificial neural network (ANN) is a widely-used model and powerful for real-time prediction and estimation of flow and sediment since early 1990s (Alp and Cigizoglu, 2007; Chau and Cheng, 2002; Cigizoglu, 2004; Dibike and Solomatine, 2001; Karunanithi et al., 1994; Kerh and Lee, 2006; Kisi, 2004b, 2004c; Melesse et al., 2011). The compact and flexible structure of ANN models enables them to be integrated with other techniques easily, which have been successfully undertaken in many studies to provide a more comprehensive model and improve model accuracy (Hassan et al., 2015; Kisi, 2010b; Kisi et al., 2006; Lafdani et al., 2013; Partal and Cigizoglu, 2008; Pramanik et al., 2011; Rajaee et al., 2010; Rajaee et al., 2011; Taormina and Chau, 2015c; Tayfur et al., 2013; Wu and Chau, 2006). The greatest superiority of ANNs over traditional modeling techniques is their ability to capture the nonlinear behavior of data without going into the details of hydrological processes. Nevertheless, in reverse, the totally implicit and physically meaningless features are also the major criticisms.

Recent research attempts to overcome the lack of physical concepts and provide insight into the unrecognized relationship within ANN models. It has great significance

particularly for the flow and sediment modeling since that they embody features of highly-nonlinear, fuzzy and time-varying in a river system. Many studies have been reported by combining ANN with conceptual model, fuzzy theory or modular methods which could provide underlying physical processes (Corzo and Solomatine, 2007; Jain and Srinivasulu, 2006; Li and Chen, 2010; Lin and Namin, 2005; Nourani and Komasi, 2013; Rezaeianzadeh et al., 2013; Song et al., 2011). However, these techniques appear to be complicated and not easily implemented, and the collection of required parameters may even not be satisfied. Other researchers have directly integrated fundamental physical principles into ANN structure to render the model more understandable (Li and Gu, 2003; Yang et al., 1998). They introduced continuity equation as activation functions into the ANN model, in which the nodes in the hidden and output layers were regarded as storage reservoirs. The introduction of fuzzy pattern-recognition into an ANN model is another practice to deal with the nonlinear and fuzzy hydrologic models (Qiu et al., 1998; Zhao and Chen, 2008). It was adopted as activation functions to connect the network input and the hidden nodes by classifying inputs into a number of categories in terms of different patterns. The number of studies which introduce physical activation functions to the ANN structure is scanty, thus, a motivation of the present study is the necessity to advance the application of ANN hybrid models for river flow and sediment load.

The selection of input variable is crucial to the neural network models (Bowden et al., 2005; Leahy et al., 2008; Partal and Cigizoglu, 2008; Xiong et al., 2015). It is a sophisticated task for the modeling of river flow and sediment, since the transport is governed by many factors including rainfall, strength of flow, sediment supply, river bed, etc. A large amount of studies concerning river flow forecasting have been undertaken from the antecedent flow records (Amiri, 2015; Hu et al., 2001; Lin et al.,

2006; Nayak et al., 2004; Wang et al., 2009a; Wei et al., 2012) while concurrently rainfall has been included as an additional input variable (Araujo et al., 2011; De Vos and Rientjes, 2005; Firat, 2008; Iritz, 1992; Noori et al., 2011; Yawson et al., 2005). The influences of upstream inputs on the downstream discharge/stage have been investigated in several studies (Badrzadeh et al., 2015; Chau and Cheng, 2002; Kerh and Lee, 2006; Pramanik and Panda, 2009; Tabari, 2016; Zounemat-Kermani et al., 2013). In addition, areal precipitation over a river basin has been pronounced as a powerful influence on river flows as well (Bao et al., 2011; Chen et al., 2006; Jena et al., 2014; Johnson et al., 1999; Rezaeianzadeh et al., 2014). In regards to the sediment estimation, rainfall, flow and sediment data are common input variables, yet, their spatial variability is seldom taken into account (Afan et al., 2015; Alp and Cigizoglu, 2007; Cigizoglu and Kisi, 2006; Cimen, 2008; Firat and Güngör, 2010; Mustafa et al., 2012; Senthil Kumar et al., 2011; Zhu et al., 2007). The need for considering factors in the upstream sections has motivated the researchers to investigate their influences on the downstream flow and sediment.

The assessment of uncertainty prediction has become a necessity for most of the modeling studies within the hydrology community (Mantovan and Todini, 2006; Montanari, 2007; Zou et al., 2002). Generally, the lack of uncertainty analysis can lead to an inflated estimation of the model reliability and applicability, particularly for a novel proposed model. The methods to estimate the uncertainty of a model output range from probabilistic forecasting method and sampling-based method, to method based on the analysis of model errors and method based on fuzzy theory (Shrestha and Solomatine, 2006). They share a common weakness however, that the uncertainty analysis is based on specific assumptions (Blasone et al., 2008; Humphrey et al., 2016; Lohani et al., 2014; Montanari and Brath, 2004; Tiwari and Chatterjee, 2010a). The

demand of directly characterizing uncertainty has triggered new research. Nowadays, uncertainty estimation based on local error and clustering (UNEEC), lower upper bound estimation (LUBE) and grey formulation are popular techniques (Alvisi et al., 2012; Khosravi et al., 2011c; Solomatine and Shrestha, 2009; Taormina and Chau, 2015a). In this regards, the LUBE is the topic of current research for uncertainty analysis, in view of its easy implementation with neural network based models and direct construction of prediction intervals (Khosravi et al., 2011a; Quan et al., 2014a, 2014c; Rana et al., 2013; Ye et al., 2016).

Undoubtedly, the hybrid models for river flow and sediment load are highly nonlinear, multi-dimensional and varying with time. Traditional optimization algorithms, e.g., the conjugate gradient algorithm and Levenberg-Marquardt (L-M) method are not applicable and may easily stick into local optima (Nourani et al., 2008; Raghuwanshi et al., 2006; Setiono and Hui, 1995). Recently, population-based optimization algorithms including differential evolution (DE), artificial bee colony (ABC) and ant colony optimization (ACO) have attracted the interest of many researchers (Afshar et al., 2009; Blum and Roli, 2003; Chiong et al., 2012; Chryssolouris et al., 1996; Karaboga et al., 2007; Konar and Bagis, 2016; Korouzhdeh et al., 2017; Kumar and Reddy, 2006; Pang et al., 2015; Qin et al., 2010; Rocca et al., 2011; Szemis et al., 2012; Uzlu et al., 2014; Vasan and Simonovic, 2010). These algorithms are very powerful in finding a global optimum since they could simultaneously search in many directions. They can also overcome the poor generalization and slow convergence speed of local optimization algorithms. Therefore, optimization with population-based algorithms has been recognized as a significant issue in this research.

1.2 Research objectives

The main objective of this thesis is to develop a feasible and reliable model respectively for river flow forecasting and suspended sediment estimation. The efficiency of LUBE method on uncertainty analysis and applications of DE, ABC and ACO algorithms as optimization techniques are also important issues. Specific objectives of this research are listed as follows:

(1) To develop a hybrid forecasting model for downstream river flow by integrating physical concepts into neural network. The effectiveness of the proposed model will be confirmed by comparing with its benchmarking models. The most appropriate input variables will be determined, and concurrently the influences of inputs on the forecasting performances will be discussed.

(2) To develop a hybrid double feedforward neural network model for suspended sediment load. The estimation performances of the proposed model will be compared with three benchmarking models and the corresponding results will be discussed. Further, the physical mechanism of the sediment transport in terms of different input combinations and river basins will be investigated.

(3) To apply the use of LUBE method in evaluating the reliability of the proposed neural network models. The ability of constructing prediction intervals (PIs) by each model will be compared and investigated.

(4) To conduct a comparative study of DE, ABC and ACO algorithms for the proposed models. The performances regarding optimization ability, convergence speed and

stability of each algorithm will be presented and discussed, and finally the most adaptive optimization algorithm will be determined.

1.3 Organization of the thesis

This thesis comprises eight chapters. The following is a brief description of each chapter.

Chapter 1 introduces the background of the research, identifies the research objectives and presents the organization of the thesis.

Chapter 2 firstly presents the modeling developments respectively for river flow forecasting and sediment load estimation. Then the input variables on the modeling as well as the selection methods are discussed. Research on the uncertainty analysis and population-based optimization algorithms is finally reviewed, in which the state-of-the-art methods and their applications are presented.

Chapter 3 develops a hybrid model for downstream river flow forecasting. The identification of the model and the process of implementation are initially presented. The performances of the proposed model on daily and hourly river flows are then presented and discussed, in which comparison with its counterpart models is involved.

Chapter 4 proposes a hybrid double feedforward neural network (HDFNN) model for suspended sediment loads estimation. The mechanism of the HDFNN model is displayed in details, derived from three existing models. The applications of the estimation models in two study cases are presented and main research findings are

discussed and concluded.

Chapter 5 determines the best input variables for the proposed models respectively in Chapter 3 and 4. In this chapter, different input combinations are formulated and their influences on model performances are presented and discussed, where single- and multi-step-ahead cases are taken into account concurrently.

Chapter 6 performs uncertainty analysis on the models proposed in Chapter 3 and 4. The LUBE method with neural network based models and the indices used for PIs evaluation are introduced. Cases studies are performed respectively for the river flow forecasting and sediment load estimation. The suitability of the proposed models is examined in comparison with their corresponding benchmark models.

Chapter 7 presents the strategies of three population-based optimization algorithms and their applications on the proposed models. The comparative studies in terms of the optimization ability, convergence speed and stability are undertaken and discussed. Advantages and limitations of each algorithm are concluded.

Chapter 8 summarizes conclusions of the research project and suggests the further work.

2 Literature Review

While the necessity and significance of data-driven models for river flow and suspended sediment have been widely recognized, the demand for model accuracy, efficiency and reliability is still a challenge. This chapter introduces the development and applications of hydrological models in the field of river flow and suspended sediment simulation. The influences of input variables on the models are reviewed concurrently. Besides, methods of uncertainty analysis aiming at confirming the reliability of models as well as applications of population-based optimization algorithms are described in this chapter.

2.1 Models for river flow forecasting

River flow forecasting has a significant importance to hydrology and hydraulic engineering owing to its use in the design and management of water resource projects. In order to control water levels/discharges and to operate water structures more efficiently, models that forecast flow discharge are desired to be of high precision and certain degree of accuracy. Different types of hydrologic models have been employed for the river flows computation, the classification of which and the corresponding representative examples are demonstrated in Figure 2.1. Mechanistic models and conceptual models have been developed to describe the behavior of the internal hydrologic processes of the river flows in many decades. They belong to knowledge-driven models, which are based upon the conservation of mass, momentum and energy on the river system. Numerical simulations such as fractional steps method and finite volume method are applied to solve differential equations in the mechanistic models (Anderson and Bates, 1994; Deng et al., 2006; Sleigh et al., 1998). They are

inflexibly programmed and need predetermined order of execution, however. Conceptual models apply simplified principles to represent the interaction of hydrological variables such as rainfall, infiltration, runoff, evapotranspiration and drainage within storage elements (Franchini and Pacciani, 1991; Nash and Sutcliffe, 1970; Wagener et al., 2004). In the recent decades, TOPMODEL (Beven et al., 1995), Xinanjiang (XAJ) model (Zhao, 1992) and probability distributed model (PDM) (Moore, 2007) are popular conceptual models.

The knowledge-driven models consider physical laws in the hydrologic processes, while data-driven models, on the contrary, directly capture the mapping between input and output variables without adopting any prior knowledge. One type of data-driven models assumes that the time series of dataset is stochastic, normal and time-invariant, in which linear/nonlinear regression models (Amemiya, 1983; Tiedeman et al., 1998), transfer function models (Kachroo, 1992; Yu et al., 1994) and linear perturbation models (Kothyari et al., 1993) are typical models. These models enjoy predefined structures but unknown parameters, thus are also termed models-based (Jain and Srinivasulu, 2004). The second type identifies models with unknown structures and parameters, and therefore, presents flexibility to some degree. Representative examples are K-nearest-neighbor (K-NN) methods (Solomatine et al., 2008; Teimouri, 2010), artificial neural network (ANN) models (Imrie et al., 2000; Kisi and Cigizoglu, 2007; Sudheer et al., 2003), fuzzy inference systems (FIS) (Cheng et al., 2005; Firat and Güngör, 2007; Pramanik and Panda, 2009), support vector machine (SVM) (Kisi and Cimen, 2011; Lin et al., 2006) and genetic programming (GP) (Guven, 2009; Savic et al., 1999; Sheta and Mahmoud, 2001). These models mainly develop on the basis of computational intelligence methods, which have proven to perform comparably to stochastic-based models (Adamowski et al., 2012; Rezaeianzadeh et al., 2014; Wang et

al., 2009a). The advantage lies on that they are available to exploit the information to model the highly-nonlinear and time-varying dynamics of the hydrological process. When compared with knowledge-driven models, the data-driven models pose practical applications refraining from the difficulty of model establishment and the uncertainty of principle representing flow processes. Therefore, the data-driven models, especially the intelligence-based methods, are gaining increasing popularity in hydrology community.

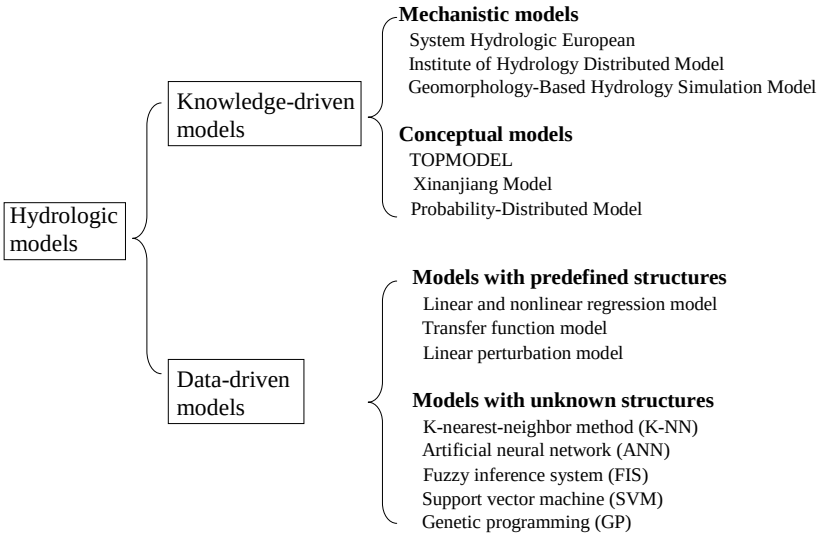

Figure 2.1 Classification of hydrologic models and the corresponding examples

Artificial neural network (ANN) model is one of the most widely-used data-driven models and powerful for real-time prediction and estimation of flow since early 1990s (Chau and Cheng, 2002; Dibike and Solomatine, 2001; Karunanithi et al., 1994; Kerh and Lee, 2006; Kisi, 2004c; Sudheer et al., 2002). It imitates the function of human brain and nervous system, and acts as an information process system. The ANN model consists of input, hidden and output layers, and corresponding nodes in each layer. The artificial nodes receive information from the previous layer, then elaborate and pass them to other artificial nodes in the next layer. Then a learning process is conducted by finding an optimal set of weights for the connection and threshold values for the nodes. The ANN models can be classified according to the direction of information flow and

processing (Govindaraju, 2000), examples of which are the multilayer perceptron (MLP) (Cannas et al., 2006; Shamseldin, 2010; Sivakumar et al., 2002), radial basis functions (RBF) (Dawson et al., 2002; Kisi, 2008b; Partal, 2008) and Bayesian neural network (BNN) (Khan and Coulibaly, 2006; Kingston et al., 2005; Lampinen and Vehtari, 2001).

The ANN models are capable of detecting any complex relationship between input and output variables due to their highly flexible structure. In particular, ANNs with one hidden layer are considered to provide enough complexity to model the nonlinear hydrologic process (De Vos and Rientjes, 2005), the structure of which is shown in Figure 2.2. By assessing the performances of ANN in comparison with traditional statistical models, it is ascertained that ANN is a remarkable alternative in river flow forecasting or flood predictions. For instance, Abrahart and See (2000) found that the ANN and autoregressive moving average (ARMA) provided similar results when applied to the river flow data in the Upper River Wye and the River Ouse; Cigizoglu (2003) demonstrated the superiority of ANN over multiple nonlinear regression (MNLR) model and autoregressive (AR) model for daily flow forecasting; the work of Kisi (2005a) showed that the ANN was able to produce better results than AR models when investigated with different inputs. Likewise, the performances of ANN models are comparably excellent when compared with other intelligence-based models such as SVM and GP models (He et al., 2014a; Kisi et al., 2012b; Londhe and Charhate, 2010; Wang et al., 2009a).

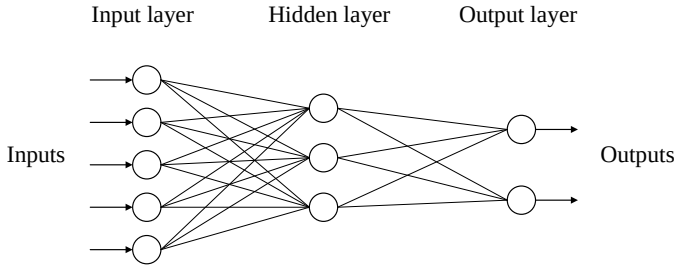

Figure 2.2 A three-layer feed-forward ANN structure

The compact and flexible structure of ANN models enables them to be integrated with other techniques easily. The ANFIS (adaptive neuro-fuzzy inference system) model is a combination of ANN and Takagi–Sugeno FIS, and has been applied successfully in river flow models (Firat and Güngör, 2007; Keskin et al., 2006; Nayak et al., 2004; Pramanik and Panda, 2009; Singh and Deo, 2007). Numerous types of data preprocessing techniques, including wavelet analysis (WA) (Cannas et al., 2006; Nourani et al., 2009; Partal, 2008), singular spectrum analysis (SSA) (Wang et al., 2015b; Wu and Chau, 2011; Wu et al., 2009a) and principal component analysis (PCA) (Hu et al., 2007; Noori et al., 2009; Okkan, 2012), have been employed to improve the performance of ANN models. The ANN models can also be combined with conceptual models for the purpose of effectively simulating the hydrological processes over various watersheds. Wilby et al. (2003) developed neural network solutions for daily discharge series simulated by a conceptual rainfall-runoff model and found a particular network that could emulate the gross behavior of the conceptual model. Song et al. (2011) integrated ANN and XAJ by considering the spatial distribution of rain-gauge stations and exploring nonlinear transformations of the runoff. Hybrid models of TOPMODEL and ANN were proposed as a simple and promising physically based rainfall-runoff model in flow modeling (Liu et al., 2013b; Xu et al., 2010). Furthermore, intelligence-based optimization algorithms integrated in ANN models to improve the generalization ability and accelerate the convergence speed are genetic algorithm (GA) (Giustolisi and Simeone, 2006; Jain and Srinivasulu, 2004; Wu and Chau, 2006), particle swarm optimization (PSO) (Chau, 2006; Taormina and Chau, 2015c; Zhang et al., 2007) and Shuffled Complex Evolution (SCE-UA) (De Vos and Rientjes, 2008; Guo et al., 2013), to name a few. In the light of literature review, it can be deduced that ANN models are quite adaptable to be integrated with fuzzy systems, data preprocessing techniques, conceptual models and optimization algorithms. The consistently satisfactory performances of these models

lead to their wide applications on hydrological forecasting. Therefore, this study pays primary attention to the hybrid ANN model for the river flow forecasting.

The ANN models yield powerful potential in forecasting issues owing to their relatively low computational demands and self-adjust to information as well. The greatest superiority of ANN models over traditional modeling techniques is their ability to capture the nonlinear behavior of data without going into the details of physical processes. Nevertheless, in reverse, the totally implicit and physically meaningless features are also the major criticisms. The confidence or reliability of networks in addressing real-word problem is difficult to build, since comprehensive decisions cannot be produced in a "black-box" ANN model (Benítez et al., 1997). Another drawback is that the structure of ANN models is unknown, thus the corresponding network design parameters (e.g., the node number in the hidden layer) are usually determined by trial-and-error methods. The over-fitting problem which may influence the generalization ability of ANN models is reported as well (Coulibaly et al., 2000; Giustolisi and Laucelli, 2005).

As a matter of fact, ANN models still provide a great deal of promises in the hydrology community despite their limitations. Recent studies make an endeavor to overcome the disadvantage of physically-meaningless and provide insight into the unrecognized relationship within ANN models. They have great significance particularly for the flow modeling since that the flows in a river system embody features of highly-nonlinear and time-varying. Jain and his coworkers demonstrated that the hidden neurons in the ANN rainfall-runoff model can be approximate various components of the hydrologic system (such as infiltration, base flow, and delayed and quick surface flow, etc.) and suggested that the distributed structure of the ANN is able to capture certain physical behavior of

the rainfall-runoff process (Jain and Srinivasulu, 2006; Jain et al., 2004; Sudheer and Jain, 2004). Integrated approaches were developed using ANN and conceptual techniques when decomposing a flow hydrograph into different segments based on physical concepts in a catchment. Ahmad and Simonovic (2005) presented the use of ANN for predicting the peak flow, timing and shape of runoff hydrograph considering causal meteorological parameters and simulating a hydrograph. Rezaeianzadeh et al. (2013) applied the standard conceptual HEC-HMS (Hydrologic Engineering Center-Hydrologic Modeling System) model and ANN for forecasting daily outflows. As mentioned, neural networks have also been combined with fuzzy theory to identify the fuzzy behavior of river flow (Li and Chen, 2010; Nourani and Komasi, 2013; Peng and Liang, 2009). Modular ANN models have been proposed since that the underlying mechanisms of low-, medium- and high-flow generation are quite different and a single global ANN model cannot distinguish the corresponding process (Minns and Hall, 1996). For instance, Zhang and Govindaraju (2000) used different modules within the network to learn subsets of the input space in predicting monthly discharges; Corzo and Solomatine (2007) addressed the baseflow separation through the use of a modular architecture that takes into account the existence of various flow regimes. The above literature reviews indicate the possibility of revealing physics in ANN models.

Combining fuzzy pattern-recognition ideas with ANN model is another alternative to handle the nonlinear and fuzzy hydrological models. Qiu et al. (1998) introduced a model termed fuzzy pattern-recognition neural network (FPNN) to forecast annual runoff at Mayadu Station in Yili River of Xijiang. A meaningful function regarding fuzzy pattern-recognition was directly employed as an activation function in the hidden layer. The fuzzy notions in this practical FPNN model were the high and low runoffs due to wet and dry seasons respectively. The goal of pattern recognition was to classify

runoffs into a number of categories to reflect the nonlinear character of the river system. Zhao and Chen (2008) applied this hybrid FPNN model in ungauged basins, considering the fuzziness of basins. Such hybrid models have demonstrated their applicability and validity in hydrology communities.

Other researchers have added physical equations into neural network to render the data-driven model more understandable. Yang et al. (1998) proposed a hydrological modeling network (HYMN) model, in which the continuity equation of flow was integrated into ANN model. The nodes in the hidden and output layers were regarded as storage reservoirs, which could provide water mass conservation and accord with the nonlinear nature. The model yielded satisfactory forecasting performances in the preliminary study of Irwell river basin at Salford University. Li and Gu (2003) further extended the HYMN model to streamflow and sediment forecasts at Jingjiang reach of the Yangtze River and Dongting Lake, China. Such HYMN model overcomes the drawback of the back-propagation (BP) algorithm in traditional ANN since the parameters have physical meanings, and the computing of output variables evolves with time. Piotrowski et al. (2007) showed that the forecasted peak concentrations of a transported pollutant in a river were more approximate when mass conservation was included. The number of studies which directly integrate physical principles into ANN structure is scanty, thus, an objective of the present study is to advance the application of ANN models with physical interpretation for river flow forecasting.

Throughout the literature, intelligence-based models for river flow forecasting show superiority over another modeling in general. In this context, the ANN might be the most popular and powerful technique to simulate the nonlinearity and non-stationarity of river flow, particularly when combined with other effective algorithms. Directly

introducing physical-meaning activation functions into ANN structure has a high potential for modeling river flow. It has an aptitude to establish relationships between complex featured variables and concurrently provide comprehensive information about the mechanisms underlying the hydrologic process.

2.2 Models for suspended sediment estimation

Sediment estimation is another important issue in a river system since it is a priority in river restoration, stable channel design, water quality assessment and other applications. In practice, it is expensive and not easy to carry out the samplings of sediment concentration. Predicting the sediment load directly from empirical or analytical equation is also an extremely difficult task since the sediment transport is complex and nonlinear. In this regard, artificial intelligence techniques have become a remarkable tool to deal with the highly nonlinear and multi-variant problem in river engineering.

Based on the mechanism by which sediment particles move, sediment transports are usually separated into two classes: bed load and suspended load. Bed load sediment is characterized by the rolling, sliding and hopping of sediment particles along the bed while suspended load sediment by the grains that are picked up off the bed and moving with the turbulent eddies in the flow. It is meaningful to classify the sediment particles since this distinction also results in different measurements of sediment data and the influencing factors on the sediment transport. Suspended sediment load (*SSL*) is of crucial importance because its presence or absence has significant impact on geomorphological and biological processes in rivers. Estimation of sediment load is a pre-work of design of many hydraulic structures. Take reservoir design for example, if sediment is underestimated it will lead to insufficient-capacity reservoirs, while

overestimation of sediment will result in over-capacity reservoirs. Additionally, suspended sediment could affect the water quality and cause irreparable damage to the environment. Thus it should be preciously forecasted and well taken measures by accurate prediction.

Mechanistic methods which fall in the category of knowledge-driven models have been consistently employed to simulate the sediment transport process (Choi et al., 2007; Jha and Bombardelli, 2011; Zhao et al., 2017). It is based on the numerical methods coded to solve partial differential equations of flow and sediment transport. They are often impractical due to long computation time and huge data required, which might be even not available or measurable (Guldal and Muftuoglu, 2001). On the other hand, data-driven models for sediment simulation rely on the analysis of time series data without explicit knowledge of the system's physical behavior. In this context, classical models such as sediment rating curve (SRC) and multi-linear regression (MLR) were widely applied at the beginning. The SRC is a relationship between the suspended sediment concentration (or yield) and river discharge. The curves are generally expressed in the form of a power equation proposed by Glysson (1987) and established by a regression analysis. Conventional SRC has been commonly used but proven to be incapable of providing sufficiently accurate estimates, since the goodness of fit implied by the relation is spurious (Demirci and Baltaci, 2013; Kisi, 2005b; McBean and Al-Nassri, 1988; Roushangar et al., 2016). Likewise, the MLR methods are not appropriate for modeling the sediment transport in view of their linear relations nature (Alp and Cigizoglu, 2007; Kisi, 2004b; Rajaee et al., 2009).

In recent decades, there has been noticed a significant rise in the number of intelligence-based methods regarding the suspended sediment, such as SVM, GP, ANN

and ANFIS. Cimen (2008) found that negative sediment estimates which were encountered in the soft computing calculations are not produced by SVM. The superiority of SVM over conventional SRC and MLR has been presented for simulating suspended sediment, where the usage of wavelet support vector machine (WSVM) and least square support vector machine (LSSVM) increases lately (Himanshu et al., 2017; Kisi, 2012; Kumar et al., 2016; Nourani and Andalib, 2015; Roushangar et al., 2016; Zounemat-Kermani et al., 2016). Aytek and Kisi (2008) utilized the GP approach for the explicit formulation of daily suspended sediment-discharge relationship and found its superiority over the SRC and MLR models. Kisi et al. (2012a) applied GP technique to daily SSL estimation in two stations in Cumberland River in U.S. and indicated that GP is superior to the ANFIS, ANN and SVM models. Gene expression programming (GEP) and linear genetic programming (LGP) are extensions of the conventional tree-based GP presented in sediment modeling (Azamathulla et al., 2013; Guven and Kisi, 2011; Kisi and Guven, 2010; Kisi and Shiri, 2012; Shamaei and Kaedi, 2016; Shiri and Kisi, 2012).

To the best of our knowledge, the ANN models are the most widely-used methods investigating the suspended sediment. The work of Cigizoglu (2004) indicated that the most frequently used ANN algorithm: multi-layer perceptron (MLP) could well capture the complex non-linear behavior of the sediment series. Melesse et al. (2011) also found that the MLP predictions for *SSL* are superior compared to predictions using MLR, MNLR and ARMA. Kisi (2004b) compared performances of MLP, RBF and GRNN (generalized regression neural network), and concluded that the MLP gave the best prediction and estimation in suspended sediment concentration (*SSC*); the RBF was best for the estimation of maximum sediment peak; the RBF and GRNN may give a better expectation for the performance in total sediment load estimation. Cigizoglu and Alp (2006) revealed the advantage of GRNN in handling the frequently encountered local

minima problem when compared with the FFBP (feedforward back propagation) in modeling sediment yield. Alp and Cigizoglu (2007) employed FFBP and RBF neural networks to estimate the daily *SSL* and presented their excellent performances in comparison with MLR. Wang et al. (2009b) concluded that both FFBP and GRNN could be successfully applied to the event-based *SSC* modeling and far superior to the classical regression. Afan et al. (2015) found that FFBP has superior performance than the RBF model in estimating daily sediment load. More reviews on the ANN applications in sediment modeling imply that they have emerged as an advanced modeling technique capable of addressing inherent non-linearity in the sediment transport processes (Ardiclioglu et al., 2007; Cigizoglu and Kisi, 2006; Joshi et al., 2016; Masoumeh and Mehdi, 2012; Mustafa et al., 2012; Singh et al., 2013).

The fuzzy nature of sediment data series necessitates the utilization of fuzzy and highly nonlinear methods to the field of sediment simulation. Fuzzy logic was accepted as good procedures in suspended sediment estimation (Demirci and Baltaci, 2013; Kisi, 2004a, 2009; Kisi et al., 2006; Kumar et al., 2012; Senthil Kumar et al., 2011). As describing the degree of 'belongingness' to a set or category by a membership number, it could better reveal the fuzzy feature of models and outperform the SRC and ANN models. Neuro-fuzzy (NF) techniques, which are combinations of artificial neural network and fuzzy logic, have been explored in the literature as well. Kisi (2005b) investigated the potential of NF in establishing streamflow-suspended sediment relationship and showed that NF gave better estimates than ANN, SRC and MLR. Its efficiency was further verified in the work of Kisi et al. (2008) when compared with RBF, FFBP, GRNN, MLR and SRC. The past decade has witnessed a few applications of NF techniques (usually refers to ANFIS) on *SSL* modeling, in which their efficiencies were compared with other novel methods or improved by combining with data

processing approaches (Azamathulla et al., 2013; Partovian et al., 2016; Rajaee et al., 2010; Shiri and Kisi, 2012; Vafakhah, 2013). In regard to the *SSC* estimation, the superiority of the NF method over RBF, GRNN, MLP and SRC was also proven by Cobaner et al. (2009). The work of Rajaee et al. (2009) demonstrated that NF and ANN models were in good agreement with the observed *SSC* values. A conjunction of wavelet analysis and NF model was employed by Mirbagheri et al. (2010) for *SSC* prediction in rivers. Shamaei and Kaedi (2016) improved the ability of NF model in *SSC* estimation by combining stacking method as a powerful machine learning technique. Malik et al. (2017) applied co-active neuro-fuzzy inference system to daily *SSC* simulation and indicated its superiority over the MLP, MLR, MNLR and SRC models. Besides, the NF was used as a benchmark model for comparison in the studies by Kisi and his coworkers when they proposed novel models for *SSC* estimation (Kisi, 2009, 2010b; Kisi and Guven, 2010; Kisi et al., 2012c; Kisi and Shiri, 2012). The capacity of NF models in handling the uncertainty and vagueness of the sediment transport has been sufficiently confirmed. However, they face a major challenge regarding the choice of membership functions (e.g., triangle, trapezoidal, bell) in different cases. Besides, it is particularly notorious for the difficulty to be integrated with optimization algorithms in the learning process. A flexible and transparent model which allows implementing the fuzzy concept in neural networks is appreciated.

As stated by Afan et al. (2016), the uninterrupted development in artificial intelligence techniques and their applications have increased competition between researchers to develop an accurate model to overcome the complexity, non-stationary and dynamism of the sediment transport behavior in rivers. In recent years, considerable attention has been paid to the data preprocessing techniques to enhance the sediment predictions. Partal and Cigizoglu (2008) employed a combined wavelet-ANN method to predict and

estimate daily suspended sediment load and showed its superiority to conventional ANN and SRC methods. Kisi (2010a) proposed neuro-wavelet technique for modeling daily suspended sediment-discharge relationship and revealed that the model produced satisfactory results. The efficiencies of WANN (wavelet-artificial neural network) and WNF (wavelet transform and neuro-fuzzy) in *SSL* prediction were demonstrated in comparison with their corresponding single model, MLR and SRC (Rajaee, 2010; Rajaee, 2011; Rajaee et al., 2010; Rajaee et al., 2011). Furthermore, Shiri and Kisi (2012) deduced that the wavelet conjunction models (combined with GEP, NF and ANN) could significantly increase the accuracy of individual models in suspended sediment estimation. Liu et al. (2013a) supported the usage of WANN model as a more accurate alternative to the FFBP in predicting *SSC* with highly nonlinear and non-stationary time series. The wavelet analysis demonstrates an aptitude to represent the signal on the base of wavelet function and reduce the data noise. Principal component analysis (PCA) and gamma test are also utilized to select independent variables in sediment transport modeling (Hassan et al., 2015; Lafdani et al., 2013; Malik et al., 2017; Pektas, 2015; Rashidi et al., 2016; Tayfur et al., 2013).

Other attempts are combining optimization algorithms with machine learning approaches to predict suspended sediment. These initial studies mainly focus on the use of genetic algorithm as an evolution optimization method. Altunkaynak (2010) proposed Geno-Kalman filtering (GKF) as a combination of GA and Kalman filtering method to predict *SSC*. Ulke et al. (2011) found that the GA improved Brooks method can be used for *SSL* prediction, and its performance was as good as traditional ANN and ANFIS applications. Adib and Jahanbakhshan (2013) determined the *SSC* in tidal rivers by GA-ANN method and found that GA performed well in optimizing the parameters in ANN model. The GA-ANN model was also applied to *SSL* prediction by Adib and

Mahmoodi (2017). Ebrahimi et al. (2013) applied GA and honey-bees mating optimization algorithms in optimizing two parameters for the sediment rating curve to estimate the *SSC*. Liu et al. (2015) used an improved genetic algorithm to optimize the RBF for measuring high *SSC* in the Yellow River and revealed that the model can effectively eliminate environmental influences and raise the measuring accuracy and stability of the system. There are a few applications of the other high-potential algorithms such as PSO, DE and ABC for the sediment models (Guo and Wang, 2010; Kisi, 2009, 2010b; Kisi et al., 2012c; Masoumeh and Mehdi, 2012).

Despite the substantial superiority of the aforementioned models, a common disadvantage is the lack of physical involvement with the inside mechanisms of sediment transport in a watershed. It is still necessary and essential to develop models with conceptual ideas to contemplate the complexity of suspended sediment behavior. Li and Gu (2003) introduced mass conservation of sediment deposition into the ANN structure, considering the time-varying nature of the sediment transport process. Lin and Namin (2005) combined a deterministic numerical model with the ANNs to predict the distribution of suspended sediments under non-uniform flow conditions. In this model, the transport process of sediments within the water column is simulated by numerically solving the advective-diffusion equation. Kumar et al. (2015a) explored the use of Tropical Rainfall Measuring Mission (TRMM-3B42) dataset for suspended sediment using neural networks. It was inferred that the TRMM-3B42 rainfall estimates can be used for the conceptual rainfall-runoff-sediment modeling. Si et al. (2017) presented a new semi-physical watershed sediment yield model for the estimation of suspended sediment in a loess region, which was composed by three modules in slope, gully, and stream phases. Notwithstanding a small amount of studies so far, the ideas that integrating physical concepts of sediment transport into data-driven models have been

identified and shown a promising future in the sediment modeling.

2.3 Input variables of hydrological models

The attempt to improve accuracy of hydrological models is a long-term topic of interest by researchers. In this regards, many issues, including identification of input variables, development of model structures and employment of optimization algorithms are encountered particularly for data-driven models. As an important step, the selection of input variables should be carefully conducted in order to facilitate the mapping of the input and output for modeling a given hydrological process. The model performances are significantly affected by the input variables imported. Negative consequence of under-specification of input variables is that the model could not simulate the potential relationship between inputs and outputs, which further results in building inaccurate models. Conversely, if the model is over-specification (i.e., the input variables are redundant), model calibration becomes more difficult due to the increased size of search space. It also adds noise to the models and thus increases uncertainty associated with model errors. In the following, the current state of input variable selections is reviewed respectively for the river flow forecasting and suspended sediment estimation models. It is imperative to appropriately identify the set of inputs for explaining the behavior of output variables after having a thorough understanding of the input influences.

2.3.1 Input variable selection for river flow forecasting models

Considering the difficulty of data measurement and scarcity of relevant data, a large amount of studies addressing river flow forecasting have been undertaken from inputs based on its own time series data (Amiri, 2015; Cheng et al., 2005; Guven, 2009; Hu et

al., 2001; Keskin et al., 2006; Lin et al., 2006; Nayak et al., 2004; Noori et al., 2009; Patel and Ramachandran, 2015; Singh and Deo, 2007; Wang et al., 2009a; Wei et al., 2012; Wei et al., 2013). It belongs to the category of univariate forecasting in the literature, which may be presented in several scales such as hourly, daily, monthly and yearly. In this regard, the challenge of input determination is to select valuable inputs from past records, which have maximum dependency with output variables and minimum redundancy for calibration (Peng et al., 2005). Input variable selection techniques have been extensively employed in this field to solve the problems, and the details will be presented in the following.

As known, the modeling entailing exogenous various apart from the flow itself is termed multivariate forecasting. In a hydrological cycle, precipitation and evaporation (or evapotranspiration) are main influencing factors on runoff, which therefore comprise the potential input variables for inflow forecasting. Rainfall has been included as an additional factor to the previous flow which is widely recognized as a necessity in model inputs (Araujo et al., 2011; Brezkova et al., 2010; De Vos and Rientjes, 2005; Dibike and Solomatine, 2001; Firat, 2008; Gopakumar et al., 2007; He et al., 2014b; Iritz, 1992; Londhe and Gavraskar, 2015; Noori et al., 2011; Rezaie-Balf et al., 2017; Wu et al., 2005; Xiong et al., 2004; Yawson et al., 2005). Some of these studies indicated that models with flows or rainfall input variable only have worse performances than those with input variables consisting of both rainfall and flow. In the meanwhile, the importance of evaporation or evapotranspiration as an input variable has been evaluated by researchers. Some of them found that they tend to be a redundant input variable (Abrahart et al., 2001; Anctil et al., 2004; Toth and Brath, 2007; Xu et al., 2009a) while others pointed out their effectiveness on flow modeling (Breinholt et al., 2012; Danh et al., 1999; Nayak et al., 2013; Wang et al., 2008a; Xu et al., 2009b;

Yawson et al., 2005). Studies that forecast flow using climate input data such as temperature, relative humidity, solar radiation and wind speed have also been witnessed in years (Alizadeh et al., 2017; Araujo et al., 2011; Chitsaz et al., 2016; Demirel et al., 2012; Noori et al., 2011; Panagoulia et al., 2017).

In the most recent decades, conceptual rainfall-runoff models are successfully employed to assess land-use impacts on hydrological processes (Anctil et al., 2004; Asadnia et al., 2014; Chang et al., 2017; Daliakopoulos and Tsanis, 2016; Ferket et al., 2010; Franchini and Pacciani, 1991; Kan et al., 2015; Moore, 2007; Moore and Bell, 2002; Rezaeianzadeh et al., 2013; Vaze et al., 2011; Wei, 2016; Wilby et al., 2003). They incorporate simplified forms of physical laws and could manifest highly nonlinear, time-variant and spatially distributed properties of rainfall-runoff process with representative parameters (Hsu et al., 1995). The interaction of hydrological variables such as rainfall, infiltration, runoff, evapotranspiration and drainage within storage elements is taken into consideration. For instance, McIntyre and Al-Qurashi (2009) applied metric-conceptual IHACRES (Identification of unit Hydrographs and Component flows from Rainfall, Evaporation and Streamflow) model to an arid catchment in Oman using a semi-distributed representation of rainfall input, considering the spatial features of rainfall and the variability and non-linearity of losses; Samain and Pauwels (2013) found that actual evapotranspiration is a crucial factor in simulating the catchment water balance and the streamflow using PDM method; Daliakopoulos and Tsanis (2016) employed Sacramento soil moisture accounting model (SAC-SMA) using monthly rainfall, streamflow and potential evapotranspiration input. The conceptual rainfall-runoff models are gaining increasing popularity since they could adequately capture the behavior of runoffs from various inputs. Furthermore, the conceptual models make allowance for the spatial variability of input variables, which is a great advance in

hydrological modeling. However, the complexity of models imposes a heavy burden on the selection of optimal input variables from potential ones.

In a river system, upstream rivers carrying water and sediment flow into downstream rivers. The discharge in downstream river, therefore, is significantly affected by that in upstream rivers (Zounemat-Kermani et al., 2013). Predicting discharge from its upstream rivers is thus meaningful and further research is required because more accurate prediction results could help to more efficiently manage the water and control flooding. A small number of studies were carried out to predict flow at the downstream end of a river by the discharge records of upstream gauging sites. Kerh and Lee (2006) employed an ANN model to forecast flood discharge at station downstream of the Kaoping River on the basis of information at stations upstream of the river. Chen et al. (2006) suggested that the upstream flow information are the key effects for modeling the flood forecast using precipitation and flow data sets of the Choshui River in central Taiwan. Pramanik and Panda (2009) estimated daily outflow from a barrage located in the downstream region of Mahanadi River basin, India, using data from the Hirakud Reservoir located upstream of the barrage, and evaluated the significance of the contribution of two upstream tributaries. Zounemat-Kermani et al. (2013) revealed that using upstream records could significantly increase the accuracy of ANN and MLR models in predicting daily stream flows by around 30%. Badrzadeh et al. (2015) conducted flood forecasting at Casino station on Richmond River, Australia from hourly rainfall and runoff data, and improved the model performances by adding an upstream river flow data (Wiangaree station) as another effective input. Tabari (2016) forecasted the daily runoff in downstream of Taleghan river using data of rain gauge and hydrometric stations which are located upstream of the river. Besides, water levels from the upstream stations have also been used in the flood forecasting (Chau and Cheng,

2002; Latt and Wittenberg, 2014; Nguyen et al., 2014; Tayfur and Moramarco, 2008; Tiwari and Chatterjee, 2010b; Wu and Chau, 2006), which also provides concrete evidence of the importance of spatial input variables in a river system.

Apart from conceptual rainfall-runoff models, the spatial variability of precipitation has been considered in recent studies. Akhtar et al. (2009) explored the use of flow length and travel time in ANN models for river flow forecasting, which was employed as a preprocessing step when incorporating the spatial precipitation information. Ozyurt and Sezer (2012) used upstream flow, precipitation and retrospective downstream flow for prediction of downstream flow in the Zamanti River Basin. More generally, mean areal precipitation over a river basin has been pronounced as a powerful influence on river flows (Bao et al., 2011; Chen et al., 2006; Jena et al., 2014; Johnson et al., 1999; Linares-Rodriguez et al., 2015; Rezaeianzadeh et al., 2014). Chen et al. (2006) found that watershed's average rainfall could provide further information and thus enhance the model accuracy. Bao et al. (2011) analyzed the influence of areal rainfall on the simulation accuracy of runoff prediction using self-similarity topography method. Rezaeianzadeh et al. (2014) showed that the area-weighted precipitation as an input to ANNs and MNLR and the spatially distributed precipitation input to ANFIS and MLR lead to more accurate predictions. Given the importance to avoid flooding on the downstream side during the rainy season, it is requisite to simulate desired outflows with appropriate upstream inputs.

There are basically two categories used to select an optimal input vector from candidates: model-based and model-free. Model-based approaches characterize the relation between input and output depending on a pre-existing model, involving methods that use *a priori* knowledge of the system, heuristic method and knowledge

extraction. The use of *a priori* knowledge of the modeled system is necessary to identify candidate inputs generally, but suffers from disadvantages of subjective and case-dependent (Campolo et al., 1999; Minns and Hall, 1996; Thirumalaiah and Deo, 2000). Heuristic method is a trial-and-error process, which chooses the most appropriate input vector by comparing the performances of calibration models with all potential inputs (Hsu et al., 1995; Jain et al., 1999; Raman and Sunilkumar, 1995; Thirumalaiah and Deo, 1998; Wilby et al., 2003). A stepwise approach is used herein to avoid total enumeration by training separate networks for each input variable (Maier et al., 1998). The heuristic methods entail training a large number of models with high computational complexity, and the problem is further exacerbated in time series studies where appropriate lags must also be chosen (Bowden et al., 2005). Sensitivity analysis is the most common method of knowledge extraction, and the difficulties in choosing a reasonable value to perturb the input by and selecting the appropriate cutoff point for input significance impede its development (Atkinson et al., 2003; Kim and Kim, 2008; Liong et al., 2000; Noh et al., 2014; Yu et al., 2006; Zhao et al., 2015).

On the other hand, model-free approaches are employed to determine the optimal input variables before model calibration. They utilize statistical measures of the degree of dependence between the candidate inputs and outputs. It is noted that the efficiency of model-free approaches is largely dependent on the statistical dependency measure used. The most common measure of statistical dependence for input variables selection is correlation, which measures only linear dependence between variables. Cross correlation (CC), autocorrelation (AC) and partial autocorrelation (PAC) are typical methods employed for input selection (Abrahart, 2003; Coulibaly et al., 2000; De Vos and Rientjes, 2005; Imrie et al., 2000; Jain et al., 2004; Sajikumar and Thandaveswara, 1999; Solomatine et al., 2008; Sudheer et al., 2002; Yaseen et al., 2016). For the case in

water resources problems, however, the input-output relationships are usually highly non-linear. Thus, non-linear statistical dependence measures are more appreciated for determining input variables so that it will not omit important inputs that are related to the output in a nonlinear fashion. Mutual information (MI) is a nonlinear statistical dependence measure that could evaluate any arbitrary dependencies between random variables (Reyhani et al., 2005). It has been used to select the subset of lags of the time series and proves to be robust due to its insensitivity to noise and data transformation (Chen et al., 2014; Tongal, 2013; Wu et al., 2010). Partial mutual information (PMI) is a partial measure of the MI criterion, having advantages of catering for input redundancy with a well-defined stopping criterion. Its efficiency in the input determination regarding water resources applications has also been verified (Bowden et al., 2005; Fernando et al., 2009; He et al., 2011; May et al., 2008). Besides, principal component analysis (Chitsaz et al., 2016; Hu et al., 2007; Ouyang, 2005), self-organizing map (Herbst et al., 2009; Ismail et al., 2012; Tiwari et al., 2013) and the gamma test (Chang et al., 2014; Noori et al., 2011; Remesan et al., 2010) are other model-free methods for hydrologic applications. These methods extricate themselves from an interaction between the potential inputs and a particular data-driven model, thus present high computational efficiency. Nevertheless, they are not always suitable input selection approaches particularly when input information could be obtained by individually irrelevant candidates.

In view of the disadvantages to each approach mentioned above, researchers advocate the use of composite methods for input variable selection. A combination of *a priori* knowledge and other methods is recommended in practice, owing to the importance of a good understanding of the hydrologic system on selecting appropriate input variables (Fernando and Jayawardena, 1998; Maier et al., 1998; Silverman and Dracup, 2000).

Other composite methods have also been reported in papers, for example, Schleiter et al. (1999) used various methods including Pearson correlation, stepwise forward regression and sensitivity analysis to determine optimal inputs for the modeling of water quality; Chang et al. (2017) utilized a combination of mutual information and cross correlation for the choice of rainfall inputs for event-based rainfall-runoff modeling. With the increasing attention paid to the selection of appropriate model inputs in the water resources applications, more effective selection methods are developed, which further facilitates the input-output mapping of the data-driven models.

2.3.2 Input variable selection for suspended sediment estimation models

The type of input variables for developing an accurate sediment model can be broadly divided into hydrologic and hydro-meteorological data. Hydrologic data mainly contains water lever, discharge and sediment, while the relevant hydro-meteorological data are commonly rainfall and temperature. These variables play different roles in the sediment transport process, and practically take effect jointly on the sediment output. Thus multivariate models have been widely reported for suspended sediment simulation, taking into account various input variables. On the other hand, discharge and sediment data have been found to be separately employed as input variables in a univariate sediment model.

Antecedent sediment records could be the most easily available input for a sediment forecasting model. Examples of papers dealing with the prediction of suspended sediment using only past sediment data are Cigizoglu (2004), Partal and Cigizoglu (2008b), Guo and Wang (2010), Rajaee et al. (2011) and Afan et al. (2015). The single input variable could not unfailingly promise the optimal input-output mapping for the

sediment models, however. For instance, Cigizoglu (2004) found that the use of upstream sediment data gives more accurate results than using the input from downstream (its own station). Afan et al. (2015) revealed that including only sediment values in the input layer was not sufficient to obtain satisfactory performance, and an additional input of flow values significantly improved the result of estimation. Recently, the importance of flow data in model inputs has been widely recognized by researchers.

Previous studies paid much attention to develop flow-sediment relationship using only the current flow data as inputs, in which the sediment rating curve (SRC) has been widely employed for the univariate estimation (Asselman, 2000; Crawford, 1991; Doomen et al., 2008; Harrington and Harrington, 2013; Horowitz, 2003; Jansson, 1997). More recently, the intelligence-based methods have been found to generally yield more accurate prediction than the traditional SRC method using the same input (Boukhrissa et al., 2013; Kisi, 2005b; Kisi et al., 2008; Liu et al., 2013a; Malik et al., 2017; Roushangar et al., 2016; Zounemat-Kermani et al., 2016). Kisi (2005b) found that the NF model whose input is the current discharge has the best results while the ANN model obtains highest accuracy from inputs containing current flow and past flow/sediment data in two study cases. Malik et al. (2017) used the gamma test to determine the best inputs from four input combinations and revealed the inferior performance by the case with only current flow input. It is a more common phenomenon indicated by the above studies that the estimations with only current flow input fail to give satisfactory performances in comparison with multivariate estimations. This is apprehensible since that single input variable could not provide sufficient information for effectively reflecting the complex features of the sediment data.

There remains the necessity of taking into account the influences of various input

variables, commonly containing the flow and sediment data on a sediment estimation model (Afan et al., 2015; Cigizoglu, 2004; Cigizoglu and Alp, 2006; Cigizoglu and Kisi, 2006; Cimen, 2008; Firat and Güngör, 2010; Guven and Kisi, 2011; Kisi, 2008a, 2009, 2010b, 2012; Mustafa et al., 2012; Rajaee et al., 2009). Cigizoglu and Kisi (2006) suggested that the current flow and one-day-ahead of sediment data should be considered as inputs in the ANN model to estimate the suspended sediment at current day using the k-fold partitioning analysis. Cigizoglu and Alp (2006) found that the additional of sediment data could not attain better accuracy when using current and past flows as input data belonging to Juniata Catchment in USA. Kisi (2008a) used correlation analysis to identify a unique input vector to the ANN model and reduce the computing time. Rajaee et al. (2009) constructed six input combinations from flow and sediment data with different lead time, and found that the ANN and NF provided the best performance from the different input combination in the Little Black River Station. Afan et al. (2015) examined different input combinations of varying antecedent value for both suspended sediment load and discharge to get more accurate results. Recently, wavelet analysis has been widely used to improve the accuracy of estimating suspended sediment from discharge and sediment data (Partal and Cigizoglu, 2008a; Rajaee, 2010; Rajaee, 2011; Rajaee et al., 2010; Rajaee et al., 2011; Shiri and Kisi, 2012).

In quite a few studies, water level is included as a suitable input variable in addition to the flow and sediment data (Jain, 2001; Lohani et al., 2007; Wang and Traore, 2009). These hydrologic data take direct effect on the sediment transport, thus could be regarded as effective input variables in the modeling. However, not only the strength of flow and sediment supply has influences on the sediment transport, but also the river bed condition and climate factors. Rainfall and temperature are two typical hydro-climatic input variables for suspended sediment estimation. Raghuwanshi et al.

(2006) estimated suspended sediment yield with different combinations of rainfall and temperature as inputs by ANN models. Alp and Cigizoglu (2007) prepared three input combinations comprising of the previous daily rainfall, sediment load and river flow data, and concluded that only rainfall data as input produced less accurate results than that with rainfall and flow inputs. Zhu et al. (2007) considered suspended sediment flux's relation with the average rainfall, temperature, rainfall intensity and water discharge, and predicted monthly sediment for the Longchuanjiang River in the Upper Yangtze Catchment, China. Some algorithms concerning statistical properties (e.g., AC, PAC and CC function of the time series) were carried out by Senthil Kumar et al. (2011) to determine input vectors from antecedent rainfall, sediment concentration, and discharge values of upstream and current stations to predict *SSC*. Lafdani et al. (2013) employed a combination of gamma test and genetic algorithm to identify the best input from stream flow and rainfall data. Demirci and Baltaci (2013) used mean water temperature instead of rainfall to predict the *SSC*. There are other studies concerning the hydro-meteorological input variables in the suspended sediment models (Cobaner et al., 2009; Goyal, 2014; Kisi and Shiri, 2012; Melesse et al., 2011; Singh et al., 2013; Vafakhah, 2013).

Hydraulic data such as bed slope and sediment particle size are common input variables for knowledge-based sediment models, but have been rarely adopted in data-driven models. Dogan (2009) studied nonlinear relations between *SSC* and independent variables: bed slope, flow discharge and sediment particle size by using ANN to predict sediment concentration. Haddadchi and Dehghani (2012) demonstrated ANN's superior performance in suspended load transport rate estimation by comparing with other nine transport formulas. They evaluated suspended load transport from hydraulic and sediment data including flow velocity, flow depth, the median size of bed material

sediments, bed load and the median size of suspended load sediments. Adib and Jahanbakhshan (2013) determined the *SSC* in tidal rivers by GA-ANN method with two different networks: the inputs of first network are distance from upstream of river, flood return period and tide return period; the inputs of second network are distance from upstream of river, flood discharge and ebb height. They found that the distance from the upstream river and flood discharge are two most effective factors on *SSC* respectively in the above two networks. Albeit satisfactory results obtained by the above studies, time series of hydraulic data are usually not available for data-driven models. Hydrologic and hydro-meteorological data, particularly the rainfall, flow and sediment data are more common and have proven important in the sediment transport. Spatial variability of the input variables should also be considered in the sediment models.

2.4 Uncertainty analysis on hydrological models

Despite the number of successful applications of data-driven models on hydrological prediction scenarios, the estimation of the uncertainties is still a practical necessity. Uncertainty associated with model errors has always been inherent in the process of hydrological modeling, which might be induced by the training data, model parameters and structure. The reliability and applicability of the models might not be ensured when only point predictions are provided. It is indicated that point predictions tend to be unreliable and questionable when the system is characterized with multi-valued data, noise and probabilistic events (Quan et al., 2014a). The probability for correct predictions could not be assessed by the point predictions, which may cause problematic decision-making when a high level of uncertainty is contained. For example, accurate and timely flood warning might not be issued if the flood forecasting is not a faithful representation of reality, where input data might be lacked, model structure might be

imperfect and probabilistic characterization of meteorological events might be included. Accordingly, a reliable hydrological prediction is essential for risk-based design and management of water systems.

Quantifying the uncertainty in hydrological predictions has attracted considerable attentions recently (Kundzewicz, 2007; Mantovan and Todini, 2006; Montanari, 2007). The preferred methods for estimating uncertainty of the model output involve probabilistic forecasting method, sampling-based method, method based on the analysis of model errors and method based on fuzzy theory (Shrestha and Solomatine, 2006). Firstly, the probabilistic forecasting method predicts the output by representing the model parameters through probability distributions defined on Bayesian theory (Krzysztofowicz, 1999). As a learning approach, the Bayesian is introduced to train the neural network models by an assumed prior distribution of parameters. The Bayesian neural network (BNN) has been employed for streamflow simulation (Humphrey et al., 2016; Zhang et al., 2009), rain-runoff modeling (Khan and Coulibaly, 2006), salinity forecasting (Kingston et al., 2005) and so on. A major drawback of the BNN is that singularity problems might be induced because of the computation of Hessian matrix. Besides, the method is only attributed to the uncertainty in the parameters, which could not address the uncertainties imported by model inputs and structure.

The second method is generally known as ensemble method, still in the limit of accommodating the uncertainties associated with model parameters. The bootstrap is one typical ensemble method, which is based on resampling with replacement of the training dataset. Bootstrap based artificial neural networks have been successfully applied in hydrological modeling to estimate confidence interval of the outputs and quantify the uncertainty of the model parameters (Abrahart, 2003; Kumar et al., 2015b;

Sharma and Tiwari, 2009; Tiwari and Chatterjee, 2010a; Tiwari and Chatterjee, 2010b). Although the bootstrap method has the advantage of avoiding complex computations of derivatives, it is very time consuming since a large number of models is required to be trained to estimate total error variances. The Monte Carlo-based approach which samples the input space with probability distribution, also belongs to the sampling-based method (Kuczera and Parent, 1998). Generalized likelihood uncertainty estimation (GLUE) is a common approach using Monte Carlo sampling in hydrologic modeling (Aronica et al., 2002; Beven and Binley, 1992; Blasone et al., 2008). Most recently, Dehghani et al. (2014) performed an uncertainty analysis based on Monte Carlo approach for streamflow drought forecasts; Golasowski et al. (2015) used Monte Carlo method to sample and run multiple rainfall-runoff simulations from the modeled input space. Despite the successful applications, the demand for probability distribution of uncertain parameters hinders a widespread use of the Monte Carlo-based method.

The method based on the analysis of model errors analyzes the statistical properties of the errors on account of assumptions for the model residuals (Montanari and Grossi, 2008). For instance, Chryssolouris et al. (1996) computed a confidence interval with the assumption of normally distributed error for the neural network; Montanari and Brath (2004) employed a meta-Gaussian approach in order to estimate the probability distribution of the model error conditioned by the simulated river flow. McInerney et al. (2017) attempted to identify the best-performing error modeling schemes from eight common approaches, including the Box-Cox transformation and the log-sinh transformation to provide the most reliable and unbiased streamflow predictions. It is found in these studies that the predictive performances of such method primarily rely on the validity and relevancy of the model error assumptions. An approach termed “uncertainty estimation based on local error and clustering”(UNEEC) is proposed by

Shrestha and Solomatine (2006), in which the probability density function of the model error is estimated via empirical distribution. That is, the distribution of residuals is conditioned on the input and possible state variables of the model, making no assumption. Besides, the UNEEC takes the joint contribution of all sources of errors into account, which is more appreciated in the decision-making process for knowing the total model uncertainty. Solomatine and Shrestha (2009) concluded that the UNEEC method generates consistent, interpretable and improved model uncertainty estimates when compared with the Monte Carlo method GLUE, the meta-Gaussian, and the QR (Quantile regression) method. Pianosi et al. (2010) extended the UNEEC method to consider parameter uncertainty when relaxing the assumption that all sources of uncertainty, including input, parameter and model structure uncertainty are explicitly manifested in the model residuals. Heretofore, UNEEC method has been used for the uncertainty estimate in hydrological modeling (Chen et al., 2015; Dogulu et al., 2014; Weerts et al., 2015; Wu and Chau, 2010).

It is indicated by Jacquin and Shamseldin (2007) that a method based on fuzzy theory is preferred since the fuzzy theory is a more appropriate tool than probability theory when the uncertainties are caused by imprecise knowledge about a real system but not from randomness. The method based on fuzzy theory uses membership functions for representing uncertainty of particular variables and characterizing vagueness in human thoughts (Kar et al., 2010; Liu et al., 2016; Lohani et al., 2014; Maskey et al., 2004). Shrestha and Solomatine (2006) developed a local uncertainty estimation model (LUEM) by partitioning the input space into different zones or clusters having similar model errors using fuzzy c-means clustering. The fuzzy c-means algorithm, which allows for each instance to belong to a cluster with some degrees, has been used as a clustering technique to reduce uncertainty in hydrological modeling (Choi and Beven, 2007).

Alvisi and Franchini (2011) developed a fuzzy neural network to produce crisp forecasts and uncertainty bands, in which the model parameters were assumed to be represented by fuzzy numbers. The output of the model was a fuzzy number accordingly, which could express the total uncertainty regarding the forecasted variable. Despite that, the calibration of the fuzzy numbers still needs to define a criterion for characterizing the membership function, and thereby limits the applicability of this method in uncertainty estimation. Alvisi et al. (2012) further improved the method by cooperating grey number theory in which the grey numbers were adopted instead of the fuzzy numbers for representing the model parameters. The grey formulation has been employed to forecast the river stage (Alvisi and Franchini, 2012) and represent the total uncertainty of a conceptual rainfall-runoff model (Alvisi et al., 2013), and is found to be valid in comparison with BNN and GLUE methods.

The majority of the aforementioned methods suffer from implementation difficulties, special assumption about data distribution and heavy computational cost. This leads to the necessity of the development of simple, premise-free and fast techniques, which could still efficiently qualify uncertainties of hydrological models. Actually, methods that directly generate prediction intervals (PIs) are more appropriate. It is not accomplished in the above methods however since they construct PIs indirectly from point predictions. The Lower Upper Bound Estimation (LUBE) method was proposed by Khosravi et al. (2011c) for construction of neural network-based PIs. It considers both coverage probability and interval width criteria with two output neurons that directly approximate the lower and upper bounds. As such, the PI construction by LUBE is as simple as point predictions, which has no assumption about the data distribution and is independent of the mean and variance values of point prediction errors. The LUBE method has proven to be more efficient in generating high quality PIs

in comparison with the delta method, Bayesian method and bootstrap method (Khosravi et al., 2011c; Quan et al., 2014a; Quan et al., 2014b; Quan et al., 2014c). Furthermore, the LUBE-PSO has proven to be efficient by Quan et al. (2014a) in which PSO is employed as optimization method to minimize the nonlinear and discontinuous cost function; Taormina and Chau (2015a) successfully combined the LUBE with Multi-Objective Fully-Informed PSO for neural network river forecasting. The LUBE method has been applied to many other fields, such as travel time prediction (Khosravi et al., 2011a), electricity load prediction (Rana et al., 2013) and flood forecast (Ye et al., 2016). The LUBE is the topic of current research for uncertainty analysis, in view of its easy implementation with neural network based models and direct construction of prediction intervals.

2.5 Population-based optimization algorithms

The training of neural network based models is a problem of nonlinear optimization, which is formulated by a machine learning algorithm (also termed training algorithm). Model parameters (e.g., weights and bias) are searched in the learning process by minimizing a loss function associated with the error between the network output and target output. The search for minimum value of loss function is implemented through the parameter space with a succession of steps, and at each step the loss will decrease by adjusting the model parameters. Based on the candidate solutions in the parameter space, the training algorithms are categorized as either local or global algorithm. A local training algorithm searches a solution within a neighboring set of candidate solutions while a global algorithm achieves optimal solution among all possible solutions.

Typical local algorithms consist of gradient descent algorithm, Newton’s method,

conjugate gradient algorithm, quasi-Newton method and Levenberg-Marquardt (L-M) method. Gradient descent is a first-order method based on gradient vector and capable of dealing with many thousand parameters (Rumelhart and Mcclelland, 1986). The Newton's method is a second-order algorithm owing to the usage of Hessian matrix (Dennis and Schnabel, 1983). The conjugate gradient method can be regarded as something intermediate between gradient descent and Newton's method, in which the training directions are conjugated with respect to the Hessian matrix, and the search produces generally faster convergence than gradient descent directions (Adeli and Hung, 1994). Quasi-Newton method is developed to solve the expensive computation of the abovementioned algorithms, by building up an approximation to the inverse Hessian (Setiono and Hui, 1995). The approximation is computed using only information on the first derivatives of the loss function. The L-M algorithm which belongs to the quasi-Newton method, is designed to work with loss functions taking the form of a sum of squared errors (Hagan and Menhaj, 1994). The convergence with the gradient vector and the Jacobian matrix renders the L-M algorithm to train fast and efficiently. The L-M algorithm is widely employed in the hydrological modeling by researchers therefore (Kisi, 2004b; Kostic et al., 2016; Nourani et al., 2008; Raghuwanshi et al., 2006; Rezaeian-Zadeh et al., 2013). However, it is inevitable that the training by L-M algorithm is easily trapped into local optima in the error surface. It is not applicable to loss functions such as the root mean squared error. Further, the L-M algorithm is not recommended when there are huge data sets in neural networks, owing to a lot of memory required in the computation of Jacobian matrix.

The disadvantages of local training algorithms such as inclination to stick into local optima and slow convergence limit their further developments. Most recently, population-based optimization algorithms that belong to global optimization techniques

have attracted the interest of researchers and witnessed tremendous applications in many fields. They are characterized by a population consisting of possible solutions to the problem, which are modified by applying different types of operators and thus moving towards a near-optimal solution area. These algorithms are very powerful in finding a global optimum since they simultaneously search in many directions by using a population of possible solutions. Generally, there are two categories of population-based optimization algorithms: evolutionary algorithms (EA) and swarm intelligence algorithms. Typical methods of EAs are the genetic algorithm (GA) and differential evolution (DE), which attempt to simulate natural evolution. The second category, swarm intelligence-based algorithms are inspired by the collective behavior of animal societies, including particle swarm optimization (PSO), artificial bee colony (ABC) and ant colony optimization (ACO).

As a branch of evolutionary algorithms, genetic algorithm (GA) was developed by Holland (1975) that seeks the solution of a problem in the form of strings of numbers by applying operators such as recombination and mutation. The search of optimization problems is thus transferred into an evolutionary process. Due to the abilities of global search and evolutionary adaptation properties, the GA has been successfully integrated into ANN models for adapting the connection weights and network architecture (Chen and Chang, 2009; Wang et al., 2011c; Young et al., 2015). It could be coupled with the L-M algorithm to speed convergence to each local minimum and used to force a broad search through parameter space (He et al., 2005; Liu et al., 2017b; Mahmoudabadi et al., 2009; Prudencio and Ludermir, 2003). Although the GA compares very favorably with local training algorithms in literatures (Arabas, 1994; Mirzaee, 2009; Riley and Ciesielski, 1998), it suffers from expensive fitness function evaluations in finding the optimal solution to complex high-dimensional, multimodal problems and tendency to

converge prematurely onto a local optimum (Boudjelaba et al., 2014; Cruz-Vega et al., 2016). Recent research of GA is focused on its modified version (Jiang et al., 2017; Tynchenko et al., 2016; Zaji and Bonakdari, 2015) and the application of NSGA II (Non-dominated Sorting Genetic Algorithm II) for multi-objective problems (Dai et al., 2017; Ercan and Goodall, 2016; Heydari et al., 2016; Nazemi et al., 2006).

Differential evolution (DE) proposed by Storn and Price (1995) is another commonly-used evolutionary algorithm. It optimizes a problem by maintaining a population of candidate solutions, and creating and searching optimal solutions with mutation, crossover and selection operations. The optimization problem is not required to be differentiable since the gradient of the problem is not used by DE algorithm. It could therefore be capable in optimization problems that are even discontinuous, noisy and change over time (Rocca et al., 2011). The DE algorithm has been applied extensively in hydrological modeling so far (Babu and Angira, 2003; Kisi, 2010b; Li et al., 2013a; Masoumeh and Mehdi, 2012; Piotrowski and Napiorkowski, 2011; Piotrowski et al., 2012; Vasan and Simonovic, 2010; Zahmatkesh et al., 2015). It has been exploited to be hybridized with local training algorithms to balance the local and global optima (Subudhi and Jena, 2008, 2011; Wang et al., 2015a). It also shows better performances than the GA in terms of convergence characteristics and computational efficiency (Bandurski and Kwedlo, 2010; Li et al., 2013a; Song et al., 2014; Wang et al., 2008b). Modified or hybrid DE algorithms have gained a growing interest in hydrological studies. For instance, Chen et al. (2009) modified the DE algorithm by adopting an evolutionary learning method to optimize the controller parameters; a hybrid mutation strategy of DE algorithm was developed by Zhao et al. (2011) where some individuals in the population searched around the current best individual and the others searched randomly with its basic mutation strategy; Yang et al. (2016a) proposed

a modified DE algorithm consisting of initialization, evaluation, reproduction and crossover to train a recurrent functional neural fuzzy network. For more complex optimization problems in the real-world applications, multi-objective differential evolution has been applied to the operation of an irrigation reservoir system (Reddy and Kumar, 2008), the calibration of reservoir flood control operation (Qin et al., 2010) and flood spreading modeling (Liu and Pender, 2013). As suggested by previous research, there is still scope to employ the DE algorithm in the training of hydrological models with promising results.

In regard to the swarm intelligence-based algorithms, the optimization is carried out by mimicking the collective behavior of a nature system, such as bird flocking, ant colonies and fish schooling. Kennedy and Eberhart (1995) proposed particle swarm optimization (PSO) that the optimal space is firstly initialized with a random population of particles and will update based on the particle's position and velocity. Its advantages, including high efficiency in finding global optimum with a large probability and high convergence rate, relative simplicity of implementation, low computation cost and multi-objective generalization, have been documented (Clerc and Kennedy, 2002; Lopez et al., 2008; Masuda and Kurihara, 2012; Qiu et al., 2014; Rana et al., 2011; Sigarchian et al., 2016). It is thus applied broadly in the field of water resources, covering the prediction of river/stream flow and water level (Buyukyildiz et al., 2014; Chau, 2006, 2007; Najafzadeh and Zahiri, 2015; Piotrowski and Napiorkowski, 2011) , reservoir operation problem (Afshar, 2009; Kumar and Reddy, 2007; Peng, 2012; Reddy and Kumar, 2007; Zhang et al., 2014), water-supply system design (Montalvo et al., 2010; Montalvo et al., 2008; Pawde et al., 2013; Suribabu and Neelakantan, 2006), optimization of rainfall-runoff models (Asadnia et al., 2014; Jakubcova et al., 2015; Liu, 2009; Lu et al., 2013; Taormina and Chau, 2015b). As probably the most commonly-used and

successful training algorithm, PSO could be used as a benchmarking technique in order to evaluate the efficiency of other algorithms. In fact, although being proven as a striking alternative to most of the local algorithms, the PSO efficacy is not always ensured with respect to other global optimization algorithms. There exist some disadvantages with the canonical or standard PSO algorithm, for instance, suffering from premature or slow convergence when the optimization problem is multimodal or high-dimensional (Huang et al., 2012; Liu et al., 2017a; Lu et al., 2008). New variants of the PSO algorithm are thus devised as growing popularity, ranging from discrete PSO (Ezzeldin et al., 2014; Li et al., 2015; Wang et al., 2011b), dynamic PSO (Chen et al., 2017; Lu et al., 2015; Rakitianskaia and Engelbrecht, 2012) and co-evolutionary PSO (He and Wang, 2007; Li et al., 2016; Yang et al., 2016b) to fully informed particle swarm (FIPS) (Mendes et al., 2004; Sebt et al., 2017; Taormina and Chau, 2015c; Tehzeeb-Ul-Hassan et al., 2012) and binary-coded PSO (Kesharaju and Nagarajah, 2017; Taormina and Chau, 2015b; Wu and Tsai, 2008).

Artificial bee colony (ABC) algorithm was introduced and popularized by Karaboga (2005) to solve numerical optimization problems initially. It simulates the intelligent foraging behavior of honey bee swarm, the colony of which consists of employed, onlooker and scout bees. Each group of bees has its duties to find or choose a food source representing a possible solution to the optimization problem, on the basis of the nectar amount of a food source corresponding to the quality of the associated solution. Although being developed quite lately, the ABC has been proven to have similar or better performances in the optimization of numerical functions and some practical models compared with the GA, PSO and DE algorithms (Agrawal and Sahu, 2015; Basu, 2013; Gozde and Taplamacioglu, 2011; Karaboga and Akay, 2009; Karaboga and Basturk, 2007; Konar and Bagis, 2016; Ning et al., 2011; Rout et al., 2012). It can be

combined with neural networks for efficiently adjusting the weights and bias in the training process (Awan et al., 2014; Ebrahimi et al., 2016; Iniesta et al., 2013; Karaboga et al., 2007; Kisi et al., 2012c; Taheri et al., 2017; Yigit et al., 2011). Besides, comparative studies between ABC and traditional back propagation algorithms in ANN applications have demonstrated the good exploration and exploitation capabilities of ABC as global optimization algorithms (Bullinaria and AlYahya, 2014; Ning et al., 2011; Ozkan et al., 2011; Uzlu et al., 2014). Similarly, the global search ability of ABC could be strengthened by improving strategies or assisted other algorithms, owing to its simple implementation and easy parallelization (Chen et al., 2016; Irani and Nasimi, 2011; Jadon et al., 2017; Jafrasteh and Fathianpour, 2017; Jia et al., 2016; Li et al., 2013b). In spite of that, our understanding and practice of ABC are far from completeness. The applications of ABC have not received sufficient attentions in hydrological fields, where the models might be characterized with complex hybrid and high sensitivity with noise. In addition, its relative performances corresponding to other population-based optimization algorithms in water resources projects remain a lack of research.

Ant colony optimization (ACO) introduced by Dorigo et al. (1996) is another example of swarm intelligence-based optimization algorithms. It is a meta-heuristic technique inspired by the foraging behavior of ants, the objective of which is to find the shortest path between food source and their colony. The search is a stochastic procedure that incorporates positive feedback of accumulated information, which may avoid local optima and search for a global optimum (Bland, 2001). The implementation of ACO in neural network training has proven to be successful (Ashena and Moghadasi, 2011; Kalinli et al., 2010; Li and Chung, 2005; Mavrovouniotis and Yang, 2015; Socha and Blum, 2007). It is also indicated in literatures that the ACO might be a viable alternative to other optimization algorithms. For instance, Maier et al. (2003) presented the

superiority of ACO over GA both in terms of computational efficiency and the ability to find near global optimal solutions, for the optimal design of water distribution systems; the analysis of well testing data using ACO was a good match to these of the GA and modified L-M methods in the work of Jung et al. (2015); Kumar and Kumar (2016) found that the ACO was capable of operating on significantly small error rates in comparison to the PSO for automated selection of biometrics fusion rules/parameters. Recently, further research and detailed applications of the ACO method have included improved ACO algorithm (Korouzhdeh et al., 2017; Nguyen et al., 2016; Pang et al., 2015), PSACO (particle swarm ant colony optimization) algorithm (Lahiri and Khalfe, 2015; Lazzus et al., 2016; Shelokar et al., 2007) and multi-objective ACO (Golding et al., 2017; He and Ma, 2014; Zhang et al., 2016). On account of our study, the potential of applying ACO to the field of hydrological forecasting is clear, e.g., see its applications in water resource problems (Afshar et al., 2009; Jalali et al., 2006; Kumar and Reddy, 2006; Maier et al., 2003; Szemis et al., 2012; Zecchin et al., 2012).

The review identifies major challenges and opportunities of the population-based optimization methods for prospective research. Their optimization ability, convergence speed and reliability will be examined and compared in this study, for the models of river flow forecasting and sediment load estimation respectively.

3 A Hybrid Neural Network for Downstream River Flow Forecasting

Starting from this chapter, population-based optimization algorithms are employed to develop neural network models for river flow and sediment load. In this first application, a hybrid model is proposed by combining physical-meaning activation function in the neural network, which is an attempt to shed a light on the process of river flowing from upstream to downstream section. In particular, the development of forecasting models is presented from traditional ANN to the proposed model, and forecasting performances are compared between different models under both daily and hourly scenarios.

3.1 Introduction

The river flow forecasting is an important factor in the design and management of water resource projects. Forecasting models of high precision and certain degree of accuracy must be designed in order to prevent flooding and operate water structures more efficiently. The downstream river flow varies depending on climate, human activities, and especially the upstream flows. The spatial factors play an important role for the formation of downstream river. Thus, to ensure the impact of upstream flows on downstream flows in a river system, forecasting model with physical mechanism of the process is necessary.

The ANN model has been accepted as a prescriptive technique for river flow forecasting. It is a “black box” which could not reflect the physical relation between the input and

output variables. Many studies have been undertaken by integrating ANN and conceptual model which are available to provide underlying physical processes (Rezaeianzadeh et al., 2013; Song et al., 2011). However, the techniques appear to be complicated and not easily implemented, and the collection of required parameters may even not be satisfied. Other researchers have directly integrated fundamental physical principles into ANN structure to render the model more understandable (Li and Gu, 2003; Piotrowski et al., 2007; Yang et al., 1998). The number of these studies is scanty, thus, a motivation of the present study is the necessity to advance the applications of hybrid neural networks for river flow forecasting. Combining fuzzy pattern-recognition ideas with ANN model is another alternative to deal with the nonlinear and fuzzy hydrological models (Li and Chen, 2010; Qiu et al., 1998; Zhao and Chen, 2008), which also will be combined in the hybrid model in view of the convenience of operation and simplicity of computation.

This chapter proposes a physics-integrated model to forecast the downstream river flow, by considering the fuzzy and time-varying features of the flows. In the following, the development of forecasting models is firstly presented, including the traditional ANN, fuzzy pattern-recognition neural network (FPNN), hydrological modeling network (HYMN) and the new hybrid neural network (HNN) model. Then the applications of the above four models to daily and hourly flow forecasts are discussed in details. The main conclusions are finally drawn in the section 3.5.

3.2 Model development

Since the forecasting of river flow is a highly nonlinear and non-stationary problem, it is customarily settled with data-driven models. The traditional ANN models could be

applied in this area because they are capable of addressing nonlinear nature of the system. The FPNN model which reflects fuzzy and nonlinear feature of the river system is viable in the field as well. The HYMN model aimed at satisfying water mass conservation in the river network is proven to be feasible when storage reservoirs are assumed. While combining the advantage of the above three models, the HNN model is finally proposed by introducing conceptual activation functions into the neural networks. The following part addresses the detailed mechanism of the four forecasting models for downstream river flow.

3.2.1 Artificial neural network model

Artificial neural network (ANN), being a data-driven model, is powerful in real-time forecasting. It imitates the function of human brain and nervous system, acting as an information process system which is composed of layers and nodes. A three-layer feed-forward neural network is the most commonly used ANN in practical applications. It consists of the input, hidden and output layers. The input layer $\{p_1, p_2, ..., p_k\}$ has k nodes, representing the data introduced to the network. The weighted sum of inputs and bias is passed with a predetermined activation function f(.) to the nodes in the hidden layer (Thirumalaiah and Deo, 1998)

$$t_i = f\left(\sum_{j=1}^{k} p_j w_{ji} + b_i\right) \tag{3.1}$$

where t_i (i=1, 2, …, s) represent nodes in the hidden layer and p_j (j=1, 2, …, k) represent nodes in the input layer. The weight parameter from the input layer to the hidden layer is denoted by w_{ji}, and b_i is the bias value. The computed nodes in the output layer are obtained by similar forward pass from nodes in the hidden layers (Thirumalaiah and Deo, 1998)

$$a_h = F(\sum_{i=1}^{s} t_i \overline{w}_{ih} + \overline{b}_h) \tag{3.2}$$

where a_h (h=1, 2, …, r) represent nodes in the output layer, and the weight parameter from the hidden layer to the output layer and bias are respectively denoted by $\overline{w}_{ih}$ and $\overline{b}_h$. For traditional ANN models, the activation functions f(.) and F(.) are usually nonlinear functions (e.g., radial basis function) and linear functions, respectively. They can reveal relations of nodes between two layers, although having no physical meanings. Based on the error between target and computed outputs, the values of weight and bias are adjusted in the learning process. By this means, an optimal set of parameters $\left\{w_{11},...,w_{ks},b_1,...,b_s,\overline{w}_{11},...,\overline{w}_{sr},\overline{b}_1,...,\overline{b}_r\right\}$ is determined. Therein, gradient descent algorithm and Levenberg-Marquardt algorithm are common optimization methods in conventional ANNs. The major drawbacks of the above methods are slow convergence rate and incapability in solving non-differential problems.

3.2.2 Fuzzy pattern-recognition neural network model

The fuzzy pattern-recognition idea could be combined with neural network by introducing a conceptual activation function. For this so-called fuzzy pattern-recognition neural network (FPNN) model, the activation function from the input layer to the hidden layer is demonstrated as follows (Qiu et al., 1998)

$$Q_i = \frac{1}{\sum_{l=1}^{C} \frac{\sum_{j=1}^{k}\left[w_{ji}\left(Q_j^{in} - M_i\right)\right]^2}{\sum_{j=1}^{k}\left[w_{ji}\left(Q_j^{in} - M_l\right)\right]^2}} \tag{3.3}$$

where Q_i (i=1, 2, …, s) represent nodes in the hidden layer and Q_j^{in} (j=1, 2, …, k) represent nodes in the input layer. The weight parameter from the input layer to the

hidden layer is denoted by w_{ji}. A model vector is denoted by $M = [M_i] = [M_l]$, that contains a number of patterns in the hidden layer. The introduction of model vector can demonstrate fuzzy pattern-recognition idea in the hidden layer, since the inputs are classified into a number of categories in terms of different patterns. The parameter *C* refers to the number of elements in the model vector as well as the number of nodes in the hidden layer (i.e., *C*=*s*). Generally, a higher value of *C* generates a higher precision for the forecasting result, since it implies that there are more categories in the hidden layer and represents a higher degree of nonlinearity. The value *C*=5 and $M = (1.0, 0.75, 0.50, 0.25, 0)$ were employed in previous study (Qiu et al., 1998), which meant that the degree of membership is 1.0 for "wet" model in wet season and 0 for "dry" model in dry season. Practically, the degrees of membership should not be linear only. However, a model vector with enough elements can circumvent this limitation. We further give a general expression for the vector *M*: if the number of the nodes in the hidden layer equals to *C* (≥2), then $M = (1.0, \frac{C\text{-}2}{C\text{-}1}, \frac{C\text{-}3}{C\text{-}1}, ..., \frac{1}{C\text{-}1}, 0)$. This would fully cover the models ranging from "wet" to "dry" season. The model vector *M* (including the value of *C*) will be determined based on the forecasting performances, which would be a trial-and-error process akin to the determination of the number of nodes in the hidden layer for classic ANNs.

The conceptual activation function of Eq. (3.3) facilitates the introduction of fuzzy ideal into a neural network. Meanwhile, the activation function of FPNN model from the hidden layer to the output layer is given by Qiu et al. (1998) as follows

$$Q_h^{out} = \frac{1}{1 + \left[\left(\sum_{i=1}^{s} Q_i \overline{w}_{ih} \right)^{-1} - 1 \right]^2} \tag{3.4}$$

where Q_h^{out} (h=1, 2, …, r) represent nodes in the output layer and $\overline{w}_{ih}$ denotes the weight parameter from the hidden layer to the output layer. Thus, in FPNN model, the parameters to be optimized are w_{ji} and $\overline{w}_{ih}$. The framework of FPNN model is depicted in Figure 3.1. The structure of three-layer feed-forward neural network is kept, however, the activation functions are replaced by Eqs. (3.3) and (3.4). When compared with traditional ANN model, the FPNN model includes a pattern-recognition concept in the algorithm, which would make the recognition effects more efficient and reveal the highly nonlinear features. It is a viable model for fuzzy-and-nonlinear systems, such as river flows with distinct seasonal features. However, in previous studies, the focuses of FPNN models are quite limited to simple implementation in stationary forecasting problems, which could be extended or combined with other algorithms.

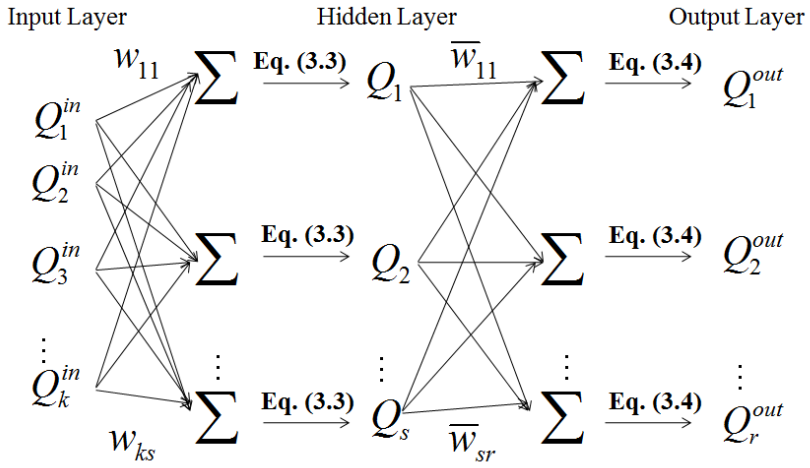

Figure 3.1 Architecture of a three-layer feed-forward FPNN model

3.2.3 Hydrological modeling neural network

The river network, in which the upstream river carries water flow into the downstream river, has to satisfy the following continuity equation (Yang et al., 1998)

$$\frac{\partial S_h}{\partial T} = \sum_{i=1}^{s} w_{ih} Q_i - Q_h \tag{3.5}$$

where S_h is water storage, Q_i and Q_h is water discharge and T is time. Meanwhile, i (1,

2, …, *s*) refers to the reservoir in a previous layer and *h* (1, 2, …, *r*) refers to the reservoir in a current layer. That is, stations in the upstream river reach are represented by reservoirs in a previous layer and stations in the downstream river reach by reservoirs in a current layer. The fraction of water from a reservoir in the previous layer entering into a reservoir in current layer is denoted by w_{ih}, which has the same meaning of the weight parameters in ANN structure. This equation implies that the rate of change of storage in the river section is determined by the difference with the source river discharge at the upstream river reach. The continuity equation is used to denote water mass conservation over the entire river system.

The discretized form of Eq. (3.5) is

$$\frac{\Delta S_h}{\Delta T} = \sum_{i=1}^{s} w_{ih} Q_i - Q_h \tag{3.6}$$

where ΔT is the time step between two layers. The water storage S_h in the current layer at time $T + \Delta T$ is determined by the following equation.

$$S_{h(T+\Delta T)} = S_{h(T)} + (\sum_{i=1}^{s} w_{ih} Q_{i(T)} - Q_{h(T)}) \times \Delta T \tag{3.7}$$

By setting $P_{h(T)} = \sum_{i=1}^{s} w_{ih} Q_{i(T)} \times \Delta T$, Eq. (3.7) in its simplified form is given by

$$S_{h(T+\Delta T)} = \lambda_{h(T)} \times (S_{h(T)} + P_{h(T)}) \tag{3.8}$$

where $\lambda_{h(T)} = 1 - \frac{Q_{h(T)} \times \Delta T}{S_{h(T)} + P_{h(T)}}$. Here λ is regarded as a recession coefficient, which is assumed to be independent of time (Yang et al., 1998). The recession coefficient represents the capability of a reservoir to absorb and store water. A higher value of the recession coefficient indicates that the reservoir can store more water. Once the storage at time $T + \Delta T$ is obtained from Eq. (3.8), the discharge in the current layer $Q_{h(T+\Delta T)}$

is evaluated as a nonlinear function of storage as follows.

$$Q_{h(T+\Delta T)} = \frac{1}{1+\exp[-(S_{h(T+\Delta T)} + P_{h(T+\Delta T)})]} \tag{3.9}$$

The nonlinear feature of the reservoir lies in the nonlinear relation between the discharge and storage of the reservoir, which can be represented by an empirical expression. The derivations of Eqs. (3.6)-(3.9) can be found in Yang et al. (1998).

The above process could be applied in the neural network to forecast downstream river discharges, which is termed hydrological modeling network (HYMN) model. In HYMN model, the river network is viewed as having the same architecture as a feed-forward neural network. The nodes in the input layer export river discharges directly. The nodes in the hidden and output layers are generalized as nonlinear reservoirs with storage capacity. They receive the outflows from the previous layer, and generate discharges after computing the entire water storage. The reservoirs in the same layer do not exchange water discharges. Thus there are two time-step parameters: ΔT_1 representing the time of flow from the input layer to the hidden layer, and ΔT_2 representing that from the hidden layer to the output layer. The water storage varies with time, denoting that the previous storage in this watershed will affect the discharge in the next time step. Consequently, large storage in wet season will result in a high discharge, exhibiting the physical phenomenon of flow in a river basin. The initial water storage $S_{h(T=0)}$ of each reservoir is prescribed before the computation. Then storage $S_{h(T+\Delta T)}$ is obtained from the initial one by every time step ΔT, which is a time-varying parameter in the model. If the storage variation term $\frac{\partial S_h}{\partial T}$ in Eq. (3.5) is neglected, HYMN model may be simplified to traditional ANN model. Three kinds of parameters are to be optimized: weight parameters w_{ji} and $\overline{w}_{ih}$, whose definitions are the same with those in ANN

model; recession coefficients λ_i and $\overline{\lambda}_h$ for the nodes in the hidden and output layers, respectively; initial storages $S_{i(T=0)}$ and $\overline{S}_{h(T=0)}$ for the nodes in the hidden and output layers, respectively. Each of them has specific meanings associated with the nonlinearity and storage-capacity of the reservoir. Accordingly, the HYMN model is applied to river system in which observed river stations are regarded as storage and nonlinear reservoirs.

3.2.4 Hybrid neural network model

The ANN, FPNN and HYMN models have proven to be efficient forecasting models for river flows in previous studies. However, there are some limitations of these models. For example, in HYMN model, the nodes in the hidden layer are assumed as storage reservoirs, which apart from being impractical, render the model complicated. Firstly, it is difficult to determine the number of reservoirs between the input and output river stations (namely the number of nodes in the hidden layer). Secondly, it is not realistic to define time step of river flow from the input layer to a virtual reservoir in the hidden layer. Finally, the recession coefficients for reservoirs in the hidden layer are physically meaningless. Above all, it is unrealistic to regard the nodes in the hidden layer as storage reservoirs.

A tentative practice is to apply Eq. (3.3) as the activation function from the nodes in the input layer to those in the hidden layer, and to consider the nodes in the output layer as storage reservoirs. This can also satisfy the mass conservation principle and represent the time-varying feature of the river network. The framework within such a hybrid neural network (HNN) model is as follows: (i) obtain the nodes in the hidden layer (i.e., determine the value of *C* by trial and error) by classifying the input variables into a

number of categories in terms of different seasons from Eq. (3.3); (ii) the sorted flows reach reservoirs in the output layer with a time step; (iii) compute the storage of reservoirs and (iv) output the discharges. In this way, the fuzzy characteristics of river flows and the time-dependent storage capacity of the observed stations are well considered. This novel model is much easier since storage reservoirs in the hidden layer are excluded. Similarly, the parameters to be optimized for the proposed HNN model are weight parameters w_{ji} and $\overline{w}_{ih}$; recession coefficients $\overline{\lambda}_h$; and initial storages $\overline{S}_{h(T=0)}$ for the reservoirs in the output layer. In the present study, the objective is to forecast the downstream river discharge. As a result, there is only one node to be considered as the reservoir in the output layer. The set of parameters to be optimized is correspondingly $\left\{w_{11},...,w_{ks},\overline{w}_{11},...,\overline{w}_{s1},\overline{\lambda},\overline{S}_{(T=0)}\right\}$.

The PSO algorithm is employed to optimize the parameters in the learning process of a HNN forecasting model, as outlined in Figure 3.2. As shown in the left segment for the framework of HNN model, the output discharge at time $T+\Delta T$ is computed from inputs (discharge *Q* and precipitation *P*) at time *T* with activation functions Eq. (3.3) and Eq. (3.9). Values of storage vary with time in particular and are obtained from previous time step values. It is worth noting that precipitation has the same nature with river flows, thus it is used as an additional input variable to be classified into different patterns in the FPNN and HNN models. The set of parameters to be optimized in HNN model include particle x_i and *D* dimensions ($D=k\times s+s\times 1+1+1$), where *D* is the number of parameters in the set, i.e., the sum of the number of w_{ji}, $\overline{w}_{ih}$, $\overline{\lambda}$ and $\overline{S}_{(T=0)}$. The process of optimizing the parameters with PSO algorithm is established in the right-hand side of Figure 3.2 and described as follows: (i) a population with randomly initialized positions and velocities within the range of parameters is generated for the

PSO algorithm; (ii) define the objective function of HNN model, which would be used to evaluate the fitness of each particle; (iii) train the proposed HNN model with parameters corresponding to the current particle and obtain the fitness value of the objective function; (iv) the velocity and position of each particle are updated in each iteration until a stopping criterion is satisfied. Consequently, an optimal set of parameters are obtained with respect to the fitness value of the objective function for the forecasting model. In such a way, a hybrid forecasting model based on continuity equation and fuzzy pattern-recognition combining with PSO algorithm is developed to forecast downstream river discharge. For the sake of fairness, PSO algorithm is employed as the optimization algorithm for the ANN, FPNN and HYMN models in this chapter. In order to verify the application of suggested models, two case study sites consisting of both daily and hourly data are explored in the following sections.

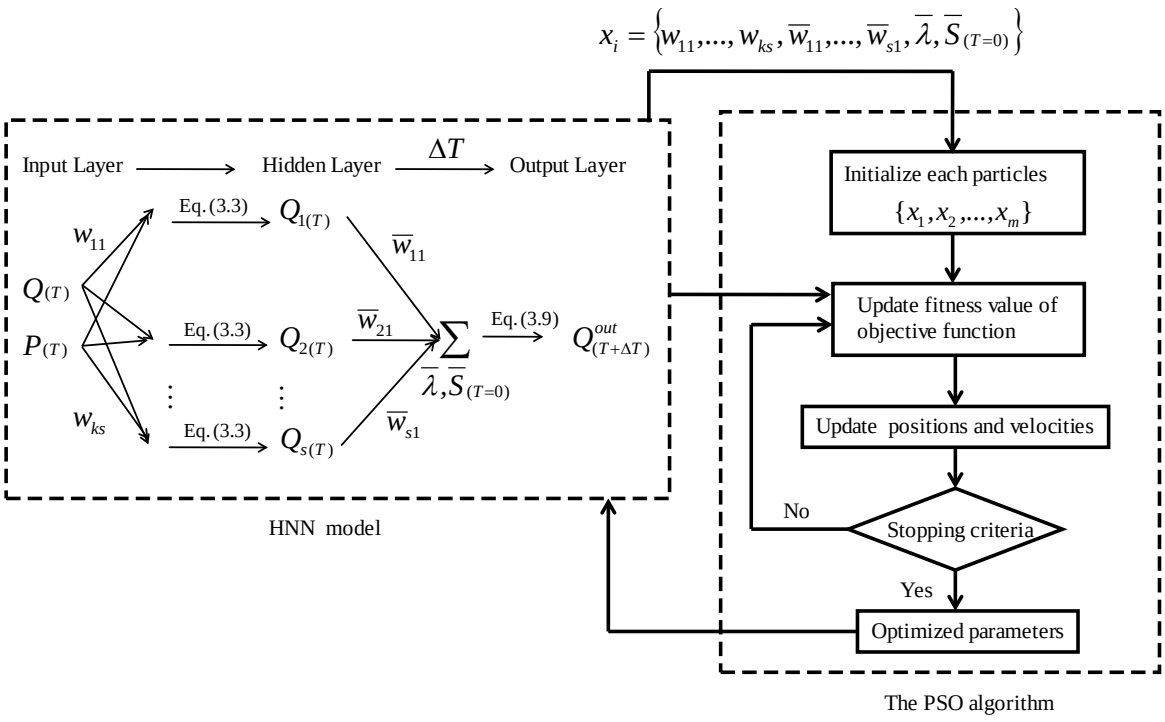

Figure 3.2 Outline of PSO algorithm for the calibration of HNN forecasting model

3.3 Daily river flow forecasting

3.3.1 Study area and data

Two study cases are performed in this study, the first of which is the Altamaha river basin located in Georgia of USA. It is a large river basin of the Atlantic coast whose drainage basin is about 36,000 km^2 in size. The Yellow River flows generally southward for 122 km, as a tributary of the Ocmulgee River. The flow of the Yellow River exhibits a seasonal behavior, which is low in dry season and high during wet months. An accurate forecasting of river discharge is vital since it allows engineers to make efficient decisions in water management and to prevent flooding. As shown in Figure 3.3, four hydrological stations along the Yellow River are marked, demonstrating their locations on the map. The corresponding station ID, name, latitude and longitude are summarized in Table 3.1. The distance from station 02207335 to 02208000 is roughly 19.3 km, and the travel time of flow between two stations is estimated to be 16 hours based on the mean velocity during flood flow condition.

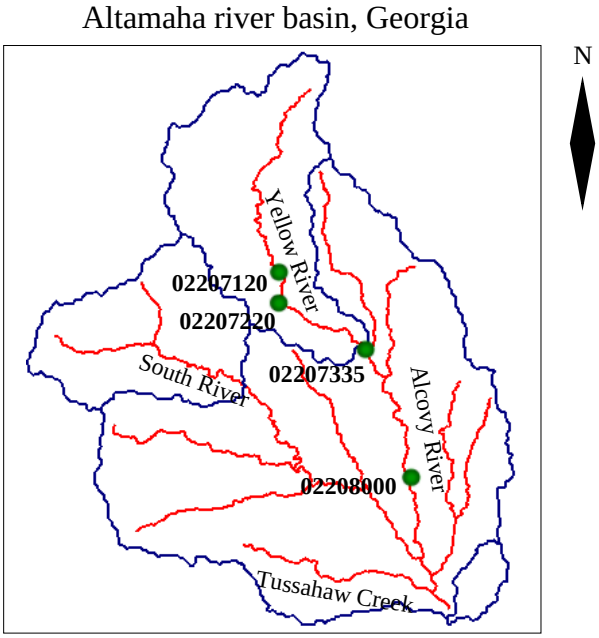

Figure 3.3 Location of gauging stations along the Yellow River in the Altamaha river basin, Georgia

Table 3.1 Stations' ID, name, location and drainage area along the Yellow River in the Altamaha river basin, Georgia

Station ID	Station name	Latitude	Longitude	Drainage area (km^2)
02207120	Yellow River at GA 124, near Lithonia	33°46'22"N	84°03'30"W	416.99
02207220	Yellow River At Pleasant Hill road, near Lithonia	33°44'01"N	84°03'43"W	551.67
02207335	Yellow River at Gees Mill road	33°40'01"N	83°56'17"W	673.40
02208000	Yellow River at Rocky Plains road	33°29'59.5"N	83°53'03"W	1108.52

The objective is to forecast the river flow at downstream station (i.e., Station 02208000) based on input variables at the upstream stations. The daily time series of discharge and precipitation in this river basin are downloaded from the USGS web server (http://waterdata.usgs.gov/ga/nwis/rt). Daily data from 1st January 2010 to 31st December 2013 were selected, and separated into training, validation and testing sets. The dataset for training stage are from 1st January 2010 to 31st December 2011, taking around 50% of all data. The data for Year 2012 are utilized for validation. During the calibration process, the training is strictly terminated at the point where the error in the validation set begins to rise. Validation is a necessary and crucial procedure to avoid over-fitting the training data (Faber and Rajko, 2007). The remaining data from 1st January 2013 to 31st December 2013 are used to assess the efficiency of the calibration and evaluate the performance of the forecasting models. The time series of discharges at Station 02208000 is demonstrated in Figure 3.4. As observed, the daily discharge varies quite irregularly, and the peak value could be as high as 199.35 m^3/s. The corresponding statistic parameters of the total and separated data are summarized in Table 3.2, in which $X_{mean,}$ S_x, X_{max}, X_{min}, C_v and C_s denote the mean, standard deviation, maximum, minimum, coefficient of variation and skewness coefficient, respectively. The training set can fully include the validation and testing data, and its statistical values are nearly

identical for the total data, which assure the availability of the selected data for the purpose of generalization and prediction.

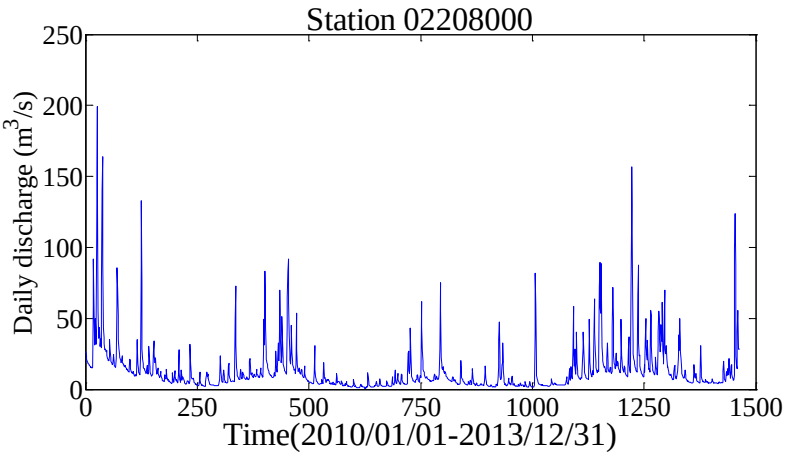

Figure 3.4 Daily discharges at Station 02208000 as output for forecasting models

Table 3.2 Pertinent information for the daily discharge data of Station 02208000

Watershed and daily datasets		Statistical parameters					
		X_{mean} (m^3/s)	S_x (m^3/s)	X_{max} (m^3/s)	X_{min} (m^3/s)	C_v	C_s
Altamaha river basin	Total data	13.23	17.63	199.35	0.71	1.33	4.28
Area: 36000 km^2	Training	13.46	18.65	199.35	0.71	1.38	4.68
Data period:	Validation	7.32	9.76	81.84	1.53	1.33	4.49
2010/01/01-2013/12/31	Testing	18.68	19.73	156.87	4.16	1.06	3.14

The downstream station 02208000 receives its flows mainly from two sources: discharge from the upstream station and precipitation at the current station. Thus two input variables are selected in the forecasting models, which are discharge at Station 02207335 and precipitation at Station 02208000. A preliminary study is carried out to examine the contribution of precipitation to the forecasting of downstream river flow by the ANN model, and the predicted discharges are demonstrated in Figure 3.5. Results show that the model with precipitation input outperforms another one with the evidence of intensively distributed dots along the ideal line. Thus precipitation is included as an indispensable input variable for this study site. As mentioned above, the nature of both upstream flow and downstream precipitation is water, and thus the connatural of these

two input variables renders the possibility of classifying the inputs into different patterns in the FPNN and HNN models.

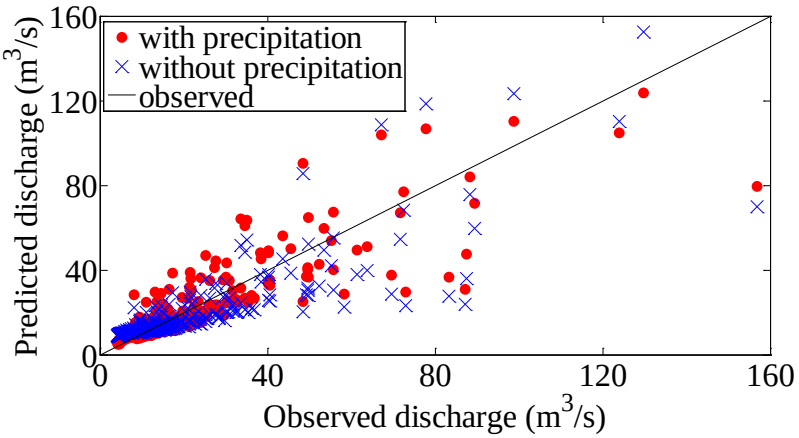

Figure 3.5 Observed and predicted daily discharges by the ANN model with different input variables in the Altamaha river basin

In addition, it is recommended to normalize each attribute in order to avoid larger data dominating smaller data. In this study, the data are scaled linearly to the range between 0.1 and 0.9 as follows (Campolo et al., 1999)

$$y_i^{'} = 0.1 + 0.8 \times \frac{y_i - y_{\min}}{y_{\max} - y_{\min}} \tag{3.10}$$

where $y_i^{'}$ is the scaled value, y_i is the original value, and $y_{\min}$, $y_{\max}$ are the minimum and maximum of the data series, respectively.

The second site locates within the central Greensboro area, North Carolina. The North Buffalo Creek is a 12.1 km tributary of Buffalo Creek in Rockbridge County in Virginia. As shown in Table 3.3 and Figure 3.6, the target station 02095500 situates the North Buffalo Creek near Greensboro, which is 10.5 km far away from the upstream station 02095271. Daily data from 1st January 2008 to 31st December 2013 are selected in this study case, which are downloaded from the USGS web server as well. As can be seen in Figure 3.7, the flows via Station 02095500 exhibit irregular, fluctuating and seasonable

behaviors. The peak value is as high as 52.95 m^3/s while around 94% of all the data is under 5.0 m^3/s. In general, the data is partitioned into training (i.e., data from Year 2008 to 2010), validation (i.e., data from 1st January 2011 to 30th June 2012) and testing set (i.e., data from 1st July 2012 to 31st December 2013). Table 3.4 provides the pertinent information for the daily data within the central Greensboro area. The training set cannot fully include the validation data, where the minimum of the training series is slightly larger than that of the validation series. However, good quality of the full dataset is fairly satisfied in terms of other pertinent information. In the same manner, two input variables: discharge at Station 02095271 and precipitation at Station 02095500 are selected for the forecasting models. The inclusion of precipitation in model inputs proves to be imperative for the present study site as well, which is concluded from the preliminary study in Figure 3.8.

Table 3.3 Stations' ID, name, location and drainage area along the North Buffalo Creek within the central Greensboro area, North Carolina

Station ID	Station name	Latitude	Longitude	Drainage area (km^2)
02095181	North Buffalo Creek at Westover Terrace at Greensboro	36°04'45"N	79°48'46"W	24.73
02095271	North Buffalo Creek at Church Street at Greensboro	36°05'52"N	79°46'57"W	36.78
02095500	North Buffalo Creek near Greensboro	36°07'14"N	79°42'29"W	96.09

Table 3.4 Pertinent information for the daily discharge data of Station 02095500

Watershed and daily datasets		Statistical parameters					
		X_{mean} (m^3/s)	S_x (m^3/s)	X_{max} (m^3/s)	X_{min} (m^3/s)	C_v	C_s
Central Greensboro area	Total data	1.50	3.33	52.95	0.23	2.22	7.82
Area: 283 km^2	Training	1.56	3.99	52.95	0.25	2.56	7.79
Data period:	Validation	1.32	2.34	26.79	0.23	1.77	6.38
2008/01/01-2013/12/31	Testing	1.57	2.66	22.57	0.27	1.69	4.41

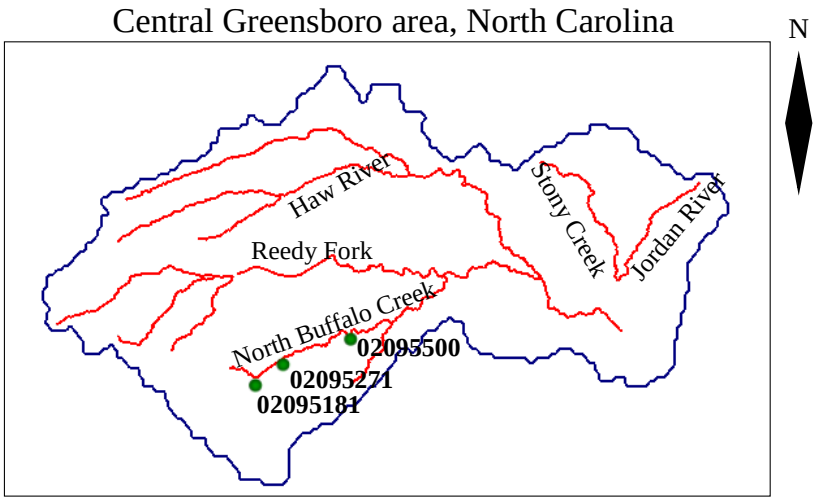

Figure 3.6 Location of gauging stations along the North Buffalo Creek within the central Greensboro area, North Carolina

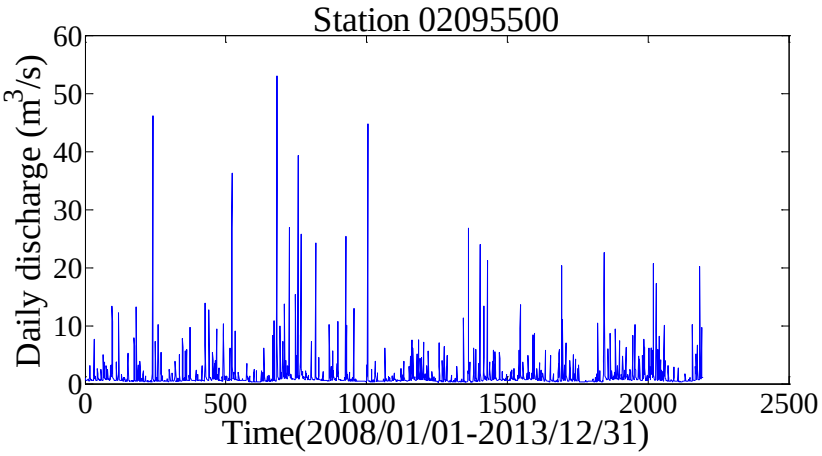

Figure 3.7 Daily discharges at Station 02095500 as output for forecasting models

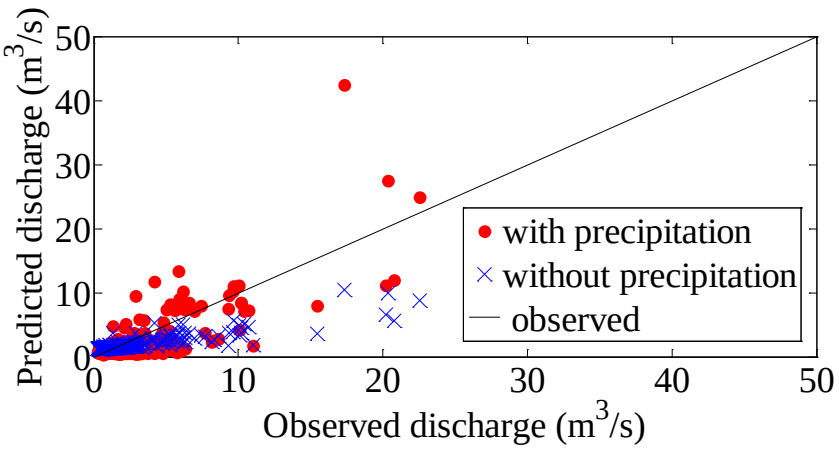

Figure 3.8 Observed and predicted daily discharges by the ANN model with different input variables within the central Greensboro area

3.3.2 Performances of forecasting models

To evaluate the performance of forecasting models, three statistical indices were used as evaluation criteria. Firstly, the RMSE (Root mean squared error) is a commonly used error index statistic which is defined as follows (Legates and McCabe, 1999)

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Q_i - \hat{Q}_i)^2} \tag{3.11}$$

It is used as the objective function in the calibration period in this study as well. Secondly, the NSEC (Nash-Sutcliffe efficiency coefficient) recommended by Nash and Sutcliffe (1970), is a normalized efficiency coefficient to assess forecasting results. It is formulated in Eq. (3.12), exhibiting the relative magnitude of the residual variance compared to the measured data variance.

$$\mathrm{NSEC} = 1 - \frac{\sum_{i=1}^{N}(Q_i - \hat{Q}_i)^2}{\sum_{i=1}^{N}(Q_i - \overline{Q})^2} \tag{3.12}$$

The statistic NSEC scales the mean squared error as well as RMSE, therefore, could reflect the performance on high values. When NSEC=1 it is a perfect fit between the forecasted discharge and the observed data; and NSEC=0 indicates that the model reaches the accuracy when the mean of the observed data is forecasted.

The NSEC is a good alternative to the coefficient of determination (Legates and McCabe, 1999) as a relative error measure since it is sensitive to differences in the observed and forecasted means and variances. The RMSE is a necessary supplement as an absolute error measure for a complete assessment of model performances. In view of checking the performance of entire data, the third statistical index known as accuracy (ACC) is employed, which is obtained from the mean relative error (De Vos and Rientjes, 2008). Its mathematical expression is presented in Eq. (3.13). Obviously, a

higher value of ACC reveals a better forecasting performance of the model and the value "1" of ACC stands for a perfect fit.

$$\mathrm{ACC} = 1 - \frac{1}{N}\sum_{i=1}^{N}\frac{\left|Q_i - \hat{Q}_i\right|}{Q_i} \tag{3.13}$$

In the above equations, Q_i and $\hat{Q}_i$ are respectively observed and forecasted discharges, $\overline{Q}$ is the mean of observed data, and N is the number of data. In the following, RMSE, NSEC and ACC are employed to evaluate the performances of various models developed.

First of all, we determined the model vector M and its corresponding value C for the proposed HNN model by comparing the fitness values during the training period. As illustrated in Figure 3.9 for the Altamaha river basin, the fitness value varies with the number of nodes in the hidden layer (i.e., value C). It decreases with the fluctuation while C increases, and attains a minimum value when C =11. The corresponding vector is $M = (1.0, 0.9, 0.8, ..., 0.1, 0)$, which is large enough to perform the nonlinear property. Figure 3.10 obtains the best C value for the HNN model within the central Greensboro area in the same manner. The number of nodes in the hidden layer for the other three forecasting models is respectively determined by comparing the fitness values in the training period similarly. The identification of the structures of the forecasting models is herein determined when the number of nodes in the hidden layer is optimized.

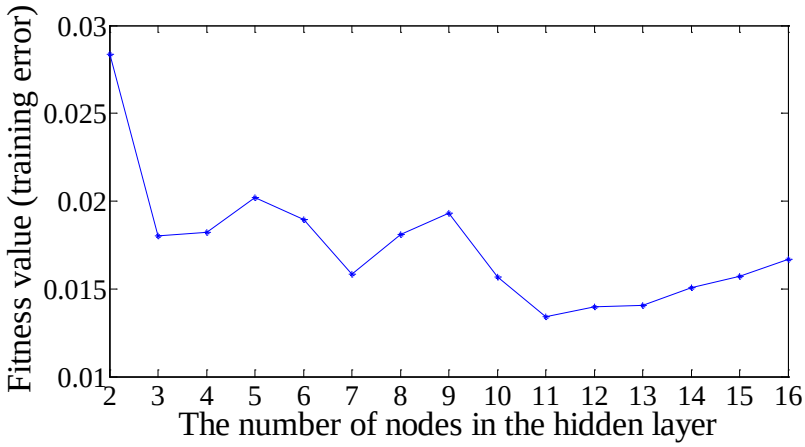

Figure 3.9 Fitness values with different numbers of nodes in the hidden layer by the HNN model in the Altamaha river basin (with daily data)

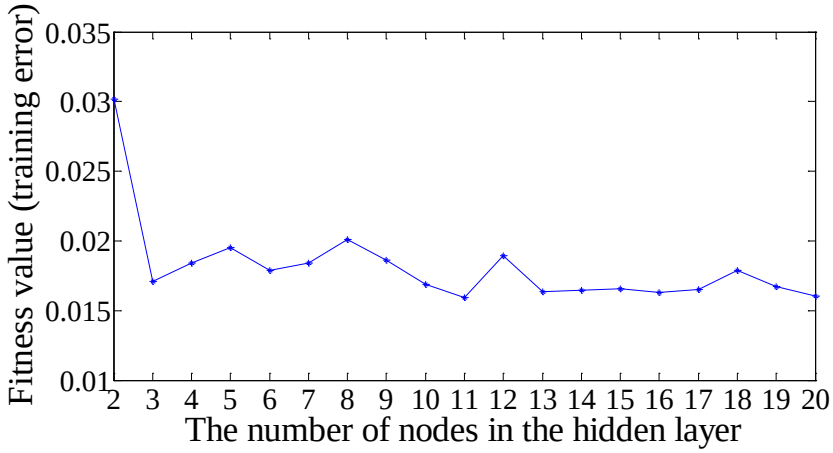

Figure 3.10 Fitness values with different numbers of nodes in the hidden layer by the HNN model within the central Greensboro area (with daily data)

As for the case in the Altamaha river basin, the lead time for ANN and FPNN model is selected as 1 day since the travel time of flow from station 02207335 to 02208000 is estimated as 16 hours. It is considered useful and necessary for practical purposes. That is, the output discharge at time T is computed from upstream discharge at time T-1 and precipitation at time T-1. Then time step ΔT is a vital parameter for both HYMN and HNN models. As demonstrated in Figure 3.2 for the HNN model, $Q^{out}_{(T+\Delta T)}$ is computed from $\{Q_{(T)}, P_{(T)}\}$, which means that the downstream flow has ΔT days delay from the inputs. That is, the inputs at day T strongly influence downstream discharge at day $T+\Delta T$. We try different time steps ΔT (1, 2, 3, 4, 5 day) for the proposed model as shown in Table 3.5 for comparison. As a similar concept with forecasting lead time, ΔT is selected

as 1 day for HYMN and HNN models with best accuracy. Accordingly, the models provide a one-day lead time forecast for the flow in Station 02208000.

Table 3.5 Training and testing performances of different ΔT values by the HNN model using daily data in the Altamaha river basin

ΔT(day)	Training			Testing		
	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC
1	**3.4326**	**0.9661**	**0.7727**	**8.3465**	**0.8210**	**0.7969**
2	5.5129	0.9126	0.6733	10.5426	0.7152	0.7758
3	6.4944	0.8789	0.6609	11.9661	0.6329	0.7451
4	6.7369	0.8698	0.6537	12.3854	0.6078	0.7505
5	6.8306	0.8663	0.6889	17.2968	0.2370	0.6832

To validate the proposed model, forecasting results were compared to three benchmarking models. The evaluation criteria RMSE, NSEC and ACC during training and testing stages by different models are provided in Table 3.6. It can be observed that the traditional ANN model seems to be adequate for the forecasting as the NSEC value during the testing period is 0.7607. However, the peak discharge computed by ANN model is about 20 m^3/s lesser than the observed value. This under-forecasting for peak discharge is intolerable for flood warning. The FPNN model provides a better forecast of the peak discharge, and is superior to ANN model in terms of the ACC value. The improvement is yet not distinct. The HYMN model performs excellently in terms of both RMSE and NSEC values. However, the ACC value is only 0.6841 in the testing period. This indicates that HYMN model is better in computing high values of discharges than ANN model and, nevertheless, cannot ensure good accuracy of all values. On the contrary, the present HNN model attains the best results for all evaluation criteria. For the training stage, there is a 56.79% reduction in RMSE value and 18.06% increase in NSEC value when compared with ANN model. The RMSE and NSEC values of HNN model are 8.3465 m^3/s and 0.8210 respectively in the testing period, with a 36.10% reduction in RMSE and 46.40% improvement in NSEC value in

comparison with FPNN model. That is, the existence of continuity equation in HNN model has made tremendous contributions on improving the forecasting performances. The increment regarding the NSEC and ACC value is 3.83% and 16.49% respectively over HYMN model in the testing stage. This reveals that the proposed model outperforms the other three benchmarking models with best generalization and forecasting ability. It may be a good alternative where an optimal forecast can be guaranteed.

Table 3.6 Training and testing performances by different models using daily data in the Altamaha river basin

Model	Number of nodes	Training			Testing			Peak Discharge (m^3/s)
		RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC	
ANN	9	7.9437	0.8183	0.7259	9.6406	0.7607	0.7289	119.83
FPNN	11	6.6239	0.8737	0.6496	13.0611	0.5608	0.7526	215.66
HYMN	6	6.0746	0.8938	0.5903	9.0272	0.7907	0.6841	136.91
HNN	11	**3.4326**	**0.9661**	**0.7727**	**8.3465**	**0.8210**	**0.7969**	218.72

To further illustrate these results, Figure 3.11 exhibits the observed and computed discharges during the testing period by the four models. It can be apparently observed that the peak value is over-forecasted by FPNN and HNN model with triangular and asterisk symbols. For the high values of computed discharges, most results from ANN and FPNN model are larger than the observed values. These two models can forecast the main data of the discharges satisfactorily, yet the computation of high values is relatively poor. Apparently, HYMN and HNN models match the observed values much better than their counterparts. In addition, we obtain the recession coefficient for reservoir in the output layer for HYMN and HNN models, which were 0.4369 and 0.4553 respectively. The coefficient computed from the two models does not vary significantly, and is thus reasonable. The reliability of the recession coefficient for reservoirs in the output layer can be assured. On the contrary, the recession coefficients

for reservoirs in the hidden layer for HYMN model vary from 0.0239 to 0.9790, revealing their uncertainty and impractical application. This offers a concrete evidence for the assumption that it is improper to regard the nodes in the hidden layer as storage reservoirs in HYMN model.

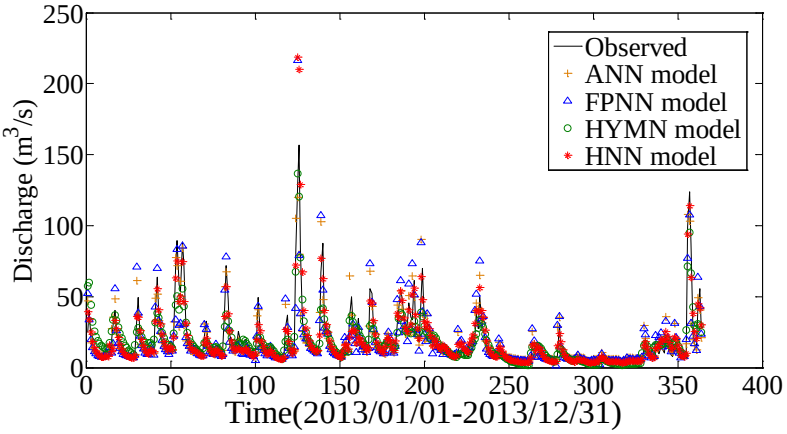

Figure 3.11 Observed and computed daily discharges during testing period by different models in the Altamaha river basin

In order to ensure wider applications of the conclusions, the performances of the study case within the central Greensboro area are demonstrated as follows. The travel time of flow from station 02095271 to 02095500 is estimated as 8 hours. Thus the one-day lead time forecast (with forecasting lead time ΔT=1 day) still gives the best accuracy, as seen in Table 3.7. However, the advantage is not distinct in comparison with the case in the Altamaha river basin. For instance, there is a 35.1% improvement in NSEC value when comparing the one- and four-day-ahead forecasts at Station 02208000, while the corresponding improvement is 18.6% at Station 02095500. That is, the flows at Station 02095500 can be forecasted with longer forecasting lead time than the flows at Station 02208000. It may be owing to the shorter distance between Station 02095500 and its nearest observed upstream station.

The comparisons of evaluation criteria and peak discharge values between the HNN and

its counterpart models are provided in Table 3.8, with one-day lead time forecast. It also reveals that the HNN model achieves the best performances among four models with best generalization and forecasting ability, since there is 2.4% and 56.8% increase in NSEC value in the training and testing period respectively comparing with ANN model. The performance of traditional ANN model is unsatisfactory with low ACC values. The relatively large difference of the RMSE values between the training and testing period also reveals the poor generalization ability of ANN model. Even worse, the peak discharge computed by ANN model is 87.7% over-forecasted, which would cause improvident preparation for flood warning. Compared with ANN model, the FPNN and HYMN models are remarkably improved in view of smaller RMSE and larger ACC values. It is noted that the ACC values attained by the HYMN model are larger than that by the HNN model, in spite of poorer RMSE and NSEC results. This indicates that HYMN model is better in obtaining good accuracy of all values but still not more superior in computing high values when compared with the HNN model in this study case. The performance of HYMN model is inconsistent with that in the Altamaha river basin. It can be explained by the feature of the time series at each observed station, as shown in Figure 3.12. There are less remarkable differences between the high and low discharges in the time series of Station 02095500 when compared with that of the Station 02208000. When the HYMN model is employed for the forecasting, the computation of the downstream flows is more consistent with small variability data series, since the storage of the reservoir is generated from its previous-step value. This may be the reason of the best accuracy by the HYMN model for the case within the central Greensboro area. The performances of RMSE and NSEC reveal the weakness in computing approximate high values, which is attributed to the absence of fuzzy pattern-recognition in the HYMN model. The peak values forecasted by the four models are higher than the observation, in which the HNN model obtains the most

approximated one. On the whole, the HNN model makes the best accuracy of the daily flow forecasting within the central Greensboro area.

In addition, the time series of observed and computed discharges within the central Greensboro area is exhibited in Figure 3.13. It can be perceived apparently that the high values are not well forecasted by the ANN model. Both HYMN and HNN models well reproduce corresponding observed discharge data and time shift between the forecasted and observed discharge does not occur. The recession coefficients for reservoir in the output layer for HYMN and HNN models are 0.1711 and 0.2066 respectively, which is approximated and acceptable. Accordingly, a conclusion can be drawn that the proposed HNN model is a considerable improvement over others and capable of producing good and approximate results. The forecasting performance is significantly enhanced as the HNN model can reflect the physical processes of the hydrological cycle within the river system more accurately with factual supports.

Table 3.7 Training and testing performances of different ΔT values by the HNN model using daily data within the central Greensboro area

ΔT(day)	Training			Testing		
	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC
1	**1.0795**	**0.9266**	**0.6662**	**1.0563**	**0.8428**	**0.6881**
2	1.3154	0.8911	0.5675	1.2146	0.7925	0.5713
3	1.3652	0.8828	0.5518	1.3826	0.7315	0.5465
4	1.2377	0.9038	0.5926	1.4356	0.7109	0.5682
5	1.3754	0.8813	0.5695	1.4452	0.7075	0.5618

Table 3.8 Training and testing performances by different models using daily data within the central Greensboro area

Model	Number of nodes	Training			Testing			Peak Discharge (m^3/s)
		RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC	
ANN	9	1.2285	0.9050	0.5089	1.8118	0.5375	0.5046	42.3619
FPNN	10	1.1127	0.9220	0.6309	1.3378	0.7479	0.6729	36.1379
HYMN	7	1.0903	0.9251	**0.7793**	1.1457	0.8151	**0.7529**	29.2781
HNN	11	**1.0795**	**0.9266**	0.6662	**1.0563**	**0.8428**	0.6881	22.8157

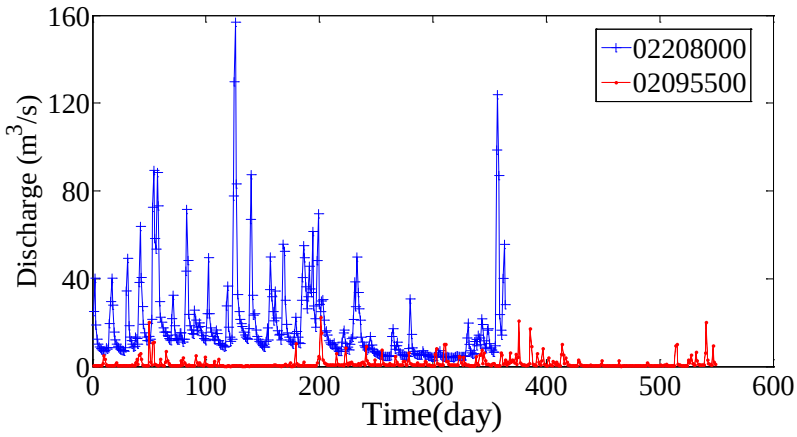

Figure 3.12 Observed daily discharges during testing period at Station 02208000 in the Altamaha river basin and 02095500 within the central Greensboro area

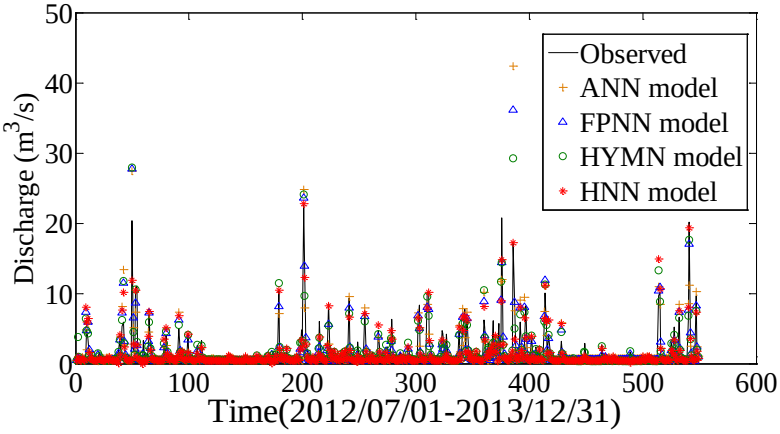

Figure 3.13 Observed and computed daily discharges during testing period by different models within the central Greensboro area

3.4 Hourly river flow forecasting

3.4.1 Study area and data

Hourly flow forecasting enjoys a realistic significance with sufficient lead time for taking appropriate flood prevention measures and rehabilitation action, which has been undertaken in many studies (Aqil et al., 2007; Dawson et al., 2002; Kang et al., 1993; Tiwari and Chatterjee, 2010a). In the section, the hourly data were selected from stations in the Altamaha river basin and the central Greensboro area as well. Two-month long hourly data are collected both, whose pertinent information is summarized in Table

3.9. The same data partition is performed here for the hourly data: the first half of the entire data as training set and the first half of the remaining data as validation set and the other half as testing set. As can be seen, the training sets of the two cases cannot fully include the validation or testing data. The *Cv* values for the observed data at Station 02095500 are commonly larger than that at Station 02208000, which reveals a wider-range spread-out data series. Figures 3.14 demonstrate the hourly time series of output variables for the first study site. As for the second case study, the hourly discharge at downstream station 02095500 as output is plotted in Figures 3.15. These more irregular and fluctuating time series indicates more difficulties in generalization. Likewise, hourly precipitation at downstream station is proven as a crucial input variable from the preliminary study.

Table 3.9 Pertinent information for the hourly discharge data of Station 02208000 and 02095500 in two watersheds

Watershed and daily datasets		Statistical parameters					
		X_{mean} (m^3/s)	S_x (m^3/s)	X_{max} (m^3/s)	X_{min} (m^3/s)	C_v	C_s
Altamaha river basin	Total data	20.25	14.36	88.91	8.78	0.71	2.96
Station 02208000	Training	26.04	17.28	88.91	14.98	0.66	2.33
Data period:	Validation	14.12	2.95	25.06	11.19	0.21	1.95
2010/03/01-2010/04/30	Testing	14.40	8.50	45.31	8.78	0.59	2.39
Central Greensboro area	Total data	2.61	7.05	68.53	0.40	2.70	6.46
Station 02095500	Training	3.27	8.98	68.53	0.51	2.75	5.52
Data period:	Validation	1.68	4.09	40.78	0.48	2.43	7.17
2013/07/01-2013/08/31	Testing	2.24	4.37	29.45	0.40	1.95	3.85

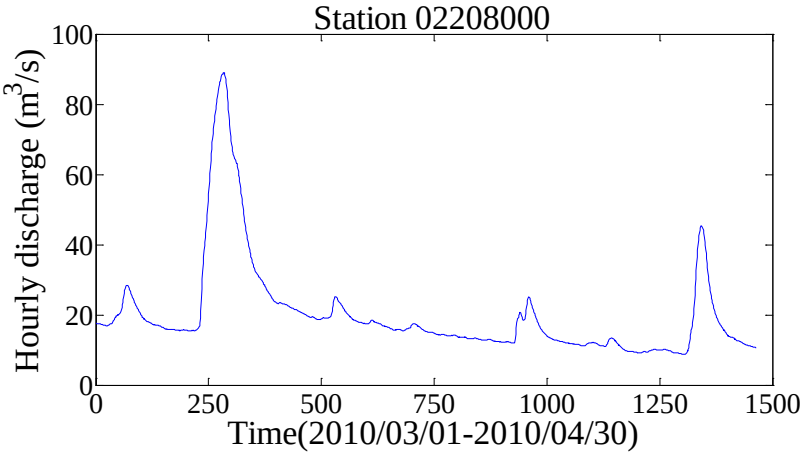

Figure 3.14 Hourly discharges at Station 02208000 as output for forecasting models

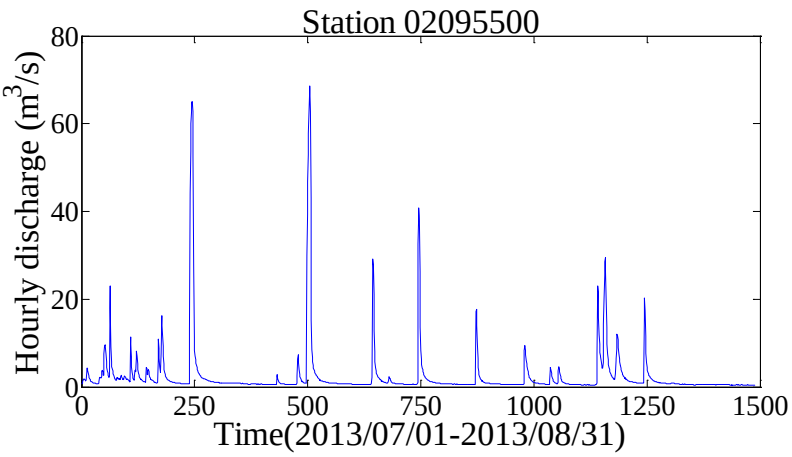

Figure 3.15 Hourly discharges at Station 02095500 as output for forecasting models

In general, the operation of neural networks entails the number of nodes in the hidden layer, which is determined by comparing the fitness value in the training period under trial tests. In particular, the results of HNN model for the two study cases are demonstrated in Figures 3.16 and 3.17, by systematically increasing the number of hidden neurons from 2 to 15. The best generalization is obtained with a same C value (i.e., C=5) for both cases, in which the corresponding vector is $M = (1.0, 0.75, 0.5, 0.25, 0)$. With increasing C values more than five, the training errors increase in a fluctuating tendency. Meantime, the model may be computationally intensive with a larger value of C. That is, it is unnecessary to classify the inputs into a large number of categories in comparison with the daily forecasting cases. This can be explained by the fact that the number of patterns is sensitive to the fuzziness of the input and output data. Since the daily time series seem to be more complicated and uncertain, the forecast should be conducted in terms of more patterns. While performing hourly forecasting, the nonlinear and fuzzy properties of flows can be fully acquired with five modes from totally wet to dry time for the study cases.

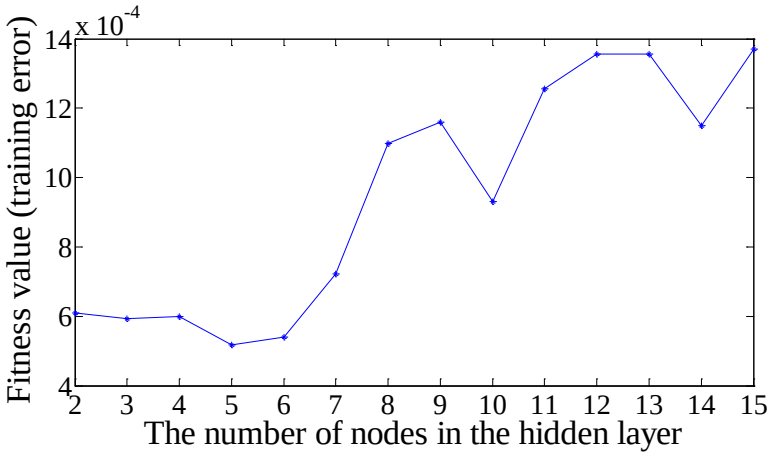

Figure 3.16 Fitness values with different numbers of nodes in the hidden layer by the HNN model in the Altamaha river basin (with hourly data)

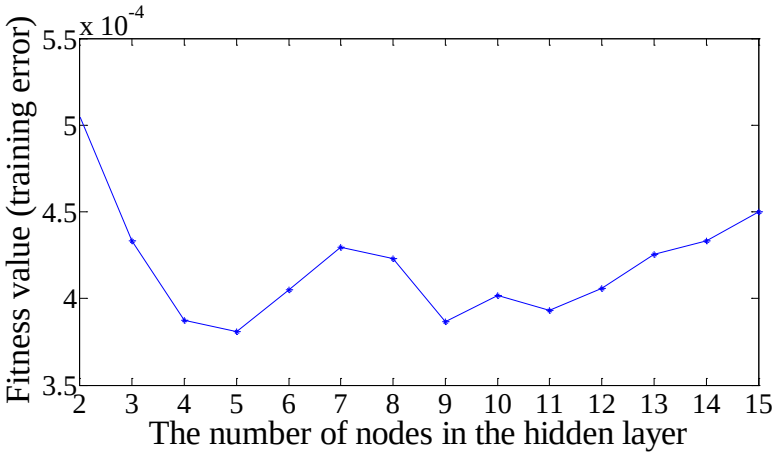

Figure 3.17 Fitness values with different numbers of nodes in the hidden layer by the HNN model within the central Greensboro area (with hourly data)

3.4.2 Performances of forecasting models

The generalization and forecasting abilities of the proposed HNN model and its validity in terms of forecast lead time are discussed with hourly data in this section. It should be recall that the time step ΔT using in HYMN and HNN models represents forecasting lead time. Table 3.10 gives the performances of HNN model with different ΔT (ΔT=1, 4, 8, 16 hours) for the study case in the Altamaha river basin. That is, the one-, four-, eight- and sixteen-hour-ahead forecasting is conducted respectively. The one-step-ahead forecasting exhibits the best results with fairly high NSEC and ACC values both in the training and testing period. The HNN model tends to be feasible in multi-step-ahead forecasting, with the evidence that the NSEC value attained by 16-hour-ahead

forecasting is as high as 0.8947 in the testing period.

Table 3.10 Training and testing performances of different ΔT values by the HNN model using hourly data in the Altamaha river basin

ΔT(hour)	Training			Testing		
	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC
1	**2.1194**	**0.9850**	**0.9403**	**1.0412**	**0.9850**	**0.9444**
4	2.6676	0.9763	0.9336	2.4826	0.9154	0.8561
8	2.5868	0.9778	0.9351	2.7455	0.8975	0.8221
16	2.7082	0.9758	0.9299	2.8114	0.8947	0.8819

With respect to the case in the Altamaha river basin, the training and testing performances by ANN, FPNN, HYMN and HNN models for 1-hour-ahead and 16-hour-ahead forecasting are presented in Table 3.11 and 3.12, respectively. It is noticeable that the ANN and FPNN models are not able to perform the 16-hour-ahead forecasting since the NSEC values in the testing period are negative. The undesirable ACC values obtained imply the unsatisfactory performances of these two models as well, which can not capture the mapping relation of the hourly forecasting. When conducting 1-hour-ahead forecasting, both ANN and FPNN models over-forecast the peak discharge with a range of 16.13% and 23.92%, and the NSEC and ACC values are comparably low. The HYMN model displays notably a good performance in computing high values with high NSEC values (i.e., 0.9789 in the training period and 0.9298 in the testing period). The HNN model generates quite low RMSE values, with a reduction of 77.99%, 77.91% and 53.79% when compared with the ANN, FPNN and HYMN models in the testing period. This reveals the advantage of HNN model in forecasting high values of the flows. The ACC values obtained by the HNN model are fairly high, presenting its ability in matching the total flow values. In addition, the recession coefficients attained by HYMN and HNN models are respectively 0.9132 and 0.9354, which is an evidence to prove the reliability of storage reservoir assumption. Therefore, the HNN model is verified as an appropriate and alternative model for the 1-hour-ahead

forecasting. The superiority of HNN model over the other three models is more distinct with the 16-hour-ahead forecasting. The assumption of storage reservoir in the HYMN model renders good RMSE and NSEC values whereas the ANN and FPNN model obtain unavailable results. Possible explanation is that the storage ability of the reservoir can be more effectively behaved regarding multi-step-ahead forecasting. Certainly, the contribution of fuzzy pattern-recognition on the performance of HNN model is promising, as there is a 2.2% reduction in RMSE value and 5.8% improvement in ACC value in comparison with the HYMN model.

Table 3.11 Training and testing performances by different models using hourly data in the Altamaha river basin (1-hour-ahead forecasting)

Model	Number of nodes	Training			Testing			Peak Discharge (m^3/s)
		RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC	
ANN	5	5.2702	0.9074	0.8843	4.7326	0.7023	0.7780	52.62
FPNN	4	4.9445	0.9185	0.8929	4.7132	0.7048	0.8858	56.15
HYMN	5	2.5076	0.9789	0.9400	2.2531	0.9298	0.8852	52.27
HNN	5	**2.1194**	**0.9850**	**0.9403**	**1.0412**	**0.9850**	**0.9444**	43.79

Table 3.12 Training and testing performances by different models using hourly data in the Altamaha river basin (16-hour-ahead forecasting)

Model	Number of nodes	Training			Testing			Peak Discharge (m^3/s)
		RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC	
ANN	5	10.0547	0.6678	0.8012	9.5912	-0.1764	0.5387	48.03
FPNN	4	9.1462	0.7251	0.8541	9.5090	-0.1564	0.5943	58.98
HYMN	5	2.9352	0.9716	0.9293	2.8748	0.8899	0.8334	39.55
HNN	5	**2.7082**	**0.9758**	**0.9299**	**2.8114**	**0.8947**	**0.8819**	43.08

The plots of observed and computed discharges by the four models with 1-hour-ahead and 16-hour-ahead forecasting depicted in Figure 3.18 and 3.19 respectively can further verify the forecasting ability of the proposed HNN model. It is worth noting that the four models make fairly obvious lagged forecasting both for the one-step-ahead and multi-step-ahead cases. The reason might be that the input data are not well manipulated

by data pre-processing techniques. Generally, the more the forecasting lead time is, the more obvious time lag it is. Another phenomenon can be observed is that the FPNN model tends to produce unexpected and discrete points in the hourly series. The probable reason may be the mismatching between these values and their mode in the model vector *M*. The absence of continuity equation from the FPNN model may result in the inconsistent points as well. Generally, the HNN model can basically fit the observed discharges in spite of slightly advanced under-forecasting for the peak value.

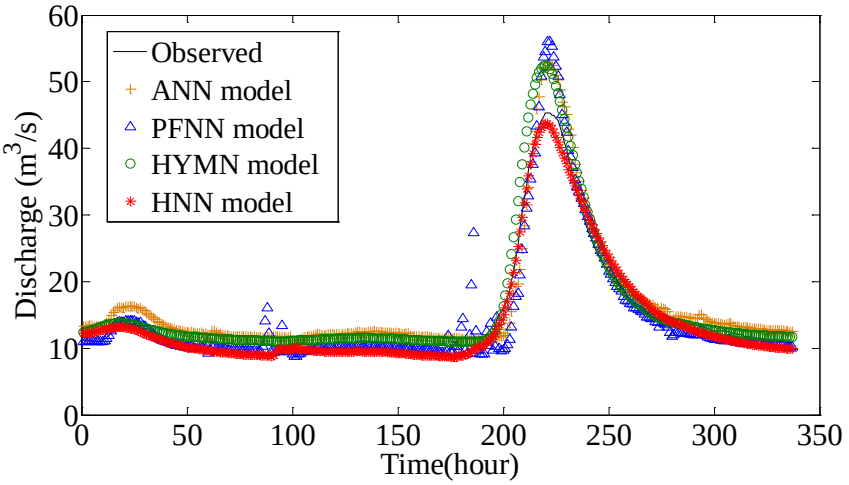

Figure 3.18 Observed and computed discharges during testing period by different models in the Altamaha river basin (1-hour-ahead forecasting)

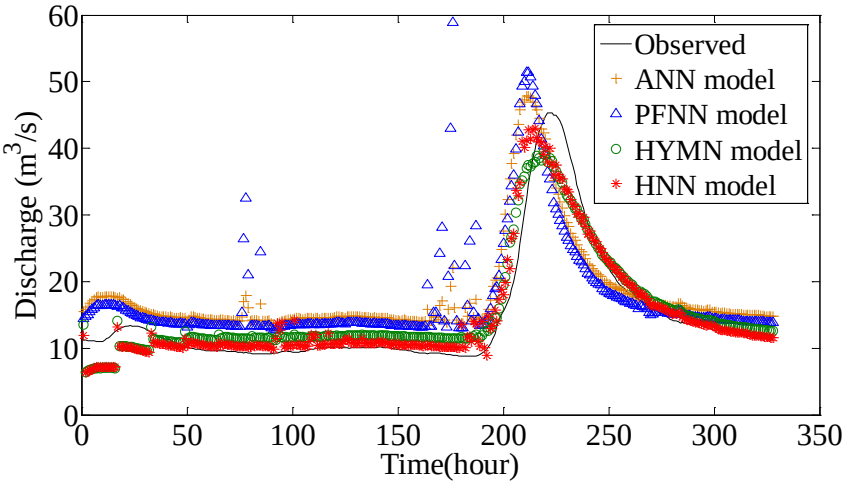

Figure 3.19 Observed and computed discharges during testing period by different models in the Altamaha river basin (16-hour-ahead forecasting)

Similar results are generated for the study case within the central Greensboro area, as shown in Tables 3.13-3.15 and Figures 3.20-3.21. The HNN model yields satisfactory

forecasting for the one-hour-ahead and four-hour-ahead cases. However, when performing more steps ahead forecasting, the NSEC and ACC values obtained are unacceptably low. This agrees with the fact that the accuracy of the forecasting is related with the lead time. Since the flow series of the North Buffalo Creek is more irregular and fluctuating than that of the previous case site, the forecasting is more sensitive to the lead time. Thus the multi-step-ahead forecasting in this area turns out to be unpromising. More evidences can be found in Table 3.15 for the case of 16-hour-ahead forecasting. Although the HNN model substantially improves the other three in terms of the RMSE and NSEC values, the ACC values are quite low and not the best one of all. When compared with the case in the Altamaha river basin in Table 3.12, the superiority of HNN model is not that distinct. That is, the HNN model is more applicable for the 16-hour-ahead forecasting in the Altamaha river basin. This may be reasoned by the fact that the recession coefficient obtained in the station 02208000 (i.e., $\overline{\lambda}$ =0.9354) is larger than that in the station 02095500 (i.e., $\overline{\lambda}$ =0.6612). In general, a larger recession coefficient implies a larger storage ability of the reservoir, which hence means the possibility to forecast more steps. As shown in Figure 3.21, many high values are over-forecasted by the ANN model and under-forecasted by the HNN model.

In spite of that, the HNN model is preferred when one-step-ahead forecasting is performed. As can be seen in Table 3.14, the HNN model obtains comparably high ACC values both in the training and testing periods. There is a reduction of 30.80%, 26.14% and 21.23% in the RMSE values when compared with the ANN, FPNN and HYMN models in the testing period. What is more, the range of over-forecasting the peak values is 19.58% by the HNN model, which is the smallest amongst the four models. Observations in Figure 3.20 reveal that the HNN model can well reproduce corresponding observed values of the last two peak valleys, but presents comparably

poor performances in the first two peak valleys. It may be due to the complicated and fluctuating time series compared with the cases in the Altamaha river basin. As a whole, the HNN model is reasonably identified for the hourly data collected in both two study cases, in particular, for multi-step-ahead forecasting.

Table 3.13 Training and testing performances of different ΔT values by the HNN model using hourly data within the central Greensboro area

ΔT(hour)	Training			Testing		
	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC
1	2.6075	0.9156	**0.7162**	**1.7310**	**0.8432**	0.7152
4	**2.5875**	**0.9170**	0.7049	1.8875	0.8140	**0.7155**
8	3.4935	0.8491	0.4556	2.5440	0.6636	0.3871
16	3.5135	0.8473	0.4357	2.5695	0.6568	0.3408

Table 3.14 Training and testing performances by different models using hourly data within the central Greensboro area (1-hour-ahead forecasting)

Model	Number of nodes	Training			Testing			Peak Discharge (m^3/s)
		RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC	
ANN	6	4.7505	0.7198	0.2992	2.5015	0.6724	0.2272	40.1450
FPNN	6	2.6425	0.9133	0.6765	2.3435	0.7125	0.6085	41.3735
HYMN	6	2.6640	0.9119	0.6371	2.1975	0.7472	0.5732	39.2650
HNN	5	**2.6075**	**0.9156**	**0.7162**	**1.7310**	**0.8432**	**0.7152**	35.2010

Table 3.15 Training and testing performances by different models using hourly data within the central Greensboro area (16-hour-ahead forecasting)

Model	Number of nodes	Training			Testing			Peak Discharge (m^3/s)
		RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC	
ANN	6	4.5350	0.7447	0.3759	3.2610	0.4434	0.2942	39.3070
FPNN	6	4.2320	0.7785	0.2861	2.9695	0.5417	0.0633	22.6890
HYMN	6	4.1025	0.7918	**0.5348**	2.7505	0.6067	**0.4972**	31.7690
HNN	5	**3.5135**	**0.8473**	0.4357	**2.5695**	**0.6568**	0.3408	32.1100

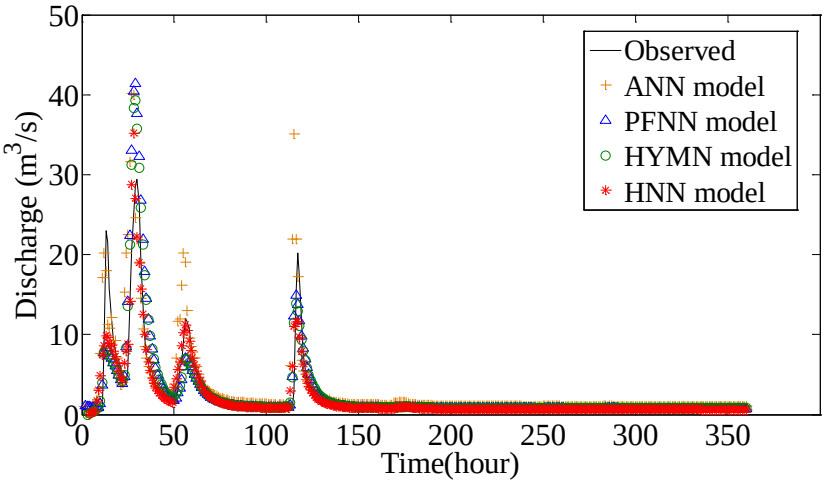

Figure 3.20 Observed and computed discharges during testing period by different models within the central Greensboro area (1-hour-ahead forecasting)

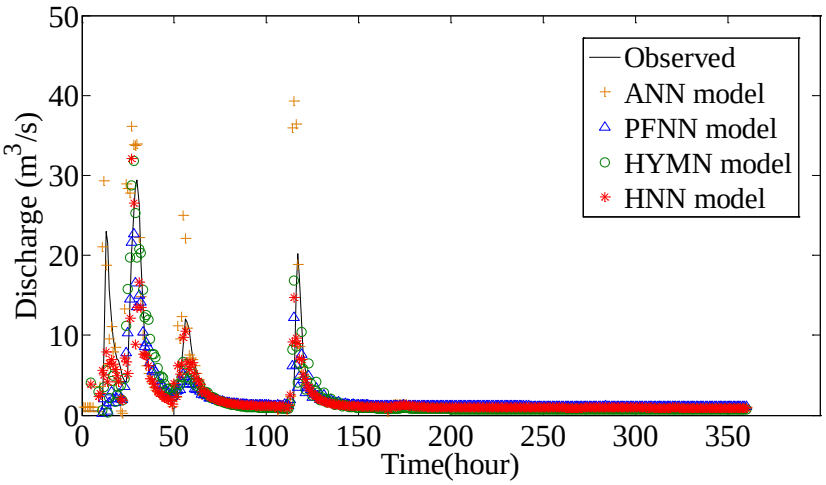

Figure 3.21 Observed and computed discharges during testing period by different models within the central Greensboro area (16-hour-ahead forecasting)

3.5 Summary

This chapter has proposed a novel hybrid model reflecting fuzzy features of river flows and nonlinear storage reservoir for downstream river flow forecasting. This model incorporates fuzzy pattern-recognition concept and continuity equation into a neural network. The integration of physical equations and data-driven model in the river system renders the forecasting more physically meaningful. To verify the capability and applicability of the proposed HNN model, three benchmarking models (namely ANN, FPNN and HYMN) are employed and two study cases are investigated under both daily and hourly scenarios. Main conclusions can be highlighted as follows:

(a) As far as forecasting high flow values are concerned, the HNN model with lowest RMSE and highest NSEC values is identified as the preferred tool among all four models, which mainly benefits from the introduction of fuzzy pattern-recognition into the hidden layer of the neural network. The HNN model appropriately fits the observed flow series with the best accuracy of the total data, since the continuity equation produces consistent computation.

(b) The reliability of assuming storage reservoir as the downstream station in the output layer of HNN model can be confirmed by the approximate recession coefficients generated by the HYMN and HNN models.

(c) Apart from the upstream flow, precipitation in the current station is employed as an important and connatural input for the downstream river flow forecasting, especially for the FPNN and HNN models with pattern classification.

(d) Since both daily and hourly forecasting are addressed in this chapter, the applicability of the HNN model shall be convincing. The travel time of flow from upstream to downstream station, the feature of time series (e.g., with more or less peak valleys, and with large or small variation) and the selected forecasting lead time will contribute to the forecasting performances of the HNN model.

(e) The superiority of HNN model over the others is more salient in the case in the Altamaha river basin because of larger storage ability of the reservoir, from the perspective of multi-step-ahead forecasting. The performance of the HNN model does not markedly deteriorate with the increase of the forecasting lead time.

The hybrid model to forecast river flow in this chapter is yet a pilot study. Since the uncertainty analysis is important for new proposed approaches, it will be undertaken as a further study. Besides, the efficiency and adaptability of the proposed model with other input variables and optimization algorithms can be investigated.

4 A Hybrid Double Feedforward Neural Network for Suspended Sediment Load Estimation

The previous chapter advocates the use of HNN as effective model for downstream river flow forecasting. The idea of integrating fuzzy pattern-recognition and continuity equation into neural network could be extended for the estimation of sediment load. Besides, a structure of double feedforward neural network is employed in the proposed model since the input variables of sediment load estimation are different in nature. Investigation will be made on two study cases with daily data:

(1) To compare the estimation performances of the proposed model with three benchmarking models;

(2) To examine the physical mechanism of the sediment load transport with different lead time and river basin.

4.1 Introduction

The estimation of suspended sediment loads (*SSL*) is required in river restoration, stable channel design and water quality assessment. It is a difficult and sophisticated task in practice, however, since the sediment transport is highly nonlinear and governed by many factors including strength of flow, sediment supply, river bed, etc. Conventional sediment rating curves (SRC) are incapable of providing sufficiently accurate estimates attributed to the misleading practice of using sediment load versus discharge (Demirci and Baltaci, 2013; McBean and Al-Nassri, 1988). Artificial intelligence techniques have proven as efficient tools in modeling sediment loads in recent years. Alp and Cigizoglu

(2007) employed two artificial neural network (ANN) models namely feedforward back propagation (FFBP) and radial basis functions (RBF) to estimate daily *SSL*. The use of support vector machine (SVM) was investigated by Cimen (2008) for *SSL* estimation in rivers. Lafdani et al. (2013) used a combination of gamma test and genetic algorithm (GT-GA) to identify the best input of SVM and ANN models for daily *SSL* prediction. These models could capture the nonlinear behavior of sediment data without going into the details of physical processes in watershed. Nevertheless, in reverse, the totally implicit and physically meaningless features are also the major criticisms. It is still imperative to develop estimation models with conceptual ideas to reflect the characteristics of sediments.

The fuzzy nature of *SSL* series necessitates the utilization of fuzzy and highly nonlinear methods for sediment simulation. A flexible and transparent model allowing implementing the fuzzy concept in activation functions is appreciated, which is inspired by the work of Qiu et al. (1998). The fuzzy pattern-recognition activation function was employed to connect the network input and the hidden nodes, and to classify inputs into a number of categories in terms of different patterns. In such a way, the fuzziness of runoff was considered with respect to the seasonal characteristic of the river system. This method offers practically significant advantage over other fuzzy-based models and is employed in this study for *SSL* estimation. In addition, the time-varying nature of sediment transport process can be considered by adding a continuity equation in the ANN structure, which has been performed by Li and Gu (2003). In their works, the nodes in the hidden/output layers were regarded as storage reservoirs, and continuity equation was satisfied when river flows from upstream to downstream sections. The spatial factors were taken into account in the sediment transport process by continuity equation, which can shed light on the effect of upstream sediment loads. Thus, this

method can build a relationship between upstream and downstream sediment loads, and is feasible and acclaimed in an *SSL* estimation model. It is preferred to completely physics-based approaches in which the detailed environmental data are generally not available and simplified assumptions are unrealistic (Kothyari et al., 1997; Kouassı et al., 2013).

Traditional multi-layer feedforward neural network (MFNN) has some drawbacks in its architecture and regularization. He (1993) proposed double parallel feedforward neural network (DPFNN), which involves a paratactic relationship between linear and nonlinear mappings. It is a parallel connection of a multi-layer feedforward neural network and a two-layer feedforward neural network. The multi-layer network used its hidden nodes to adjust the nonlinear mapping; and the two-layer network can give high learning speed for linear solution (He 1994). It was demonstrated that DPFNN has faster convergence speed and better generalization capability than MFNN (Wang et al., 2011a; Zhong and Ding, 2005). The DPFNN model has been used for hyperspectral data classification (He and Huang, 2005), concentration estimation of gas mixture (Zhao et al., 2010) and water diversion demand estimate (Khan et al., 2014), and proven to be a promising method for regression and prediction.

The purpose of this chapter is to develop a novel estimation model with a combination of fuzzy pattern-recognition, continuity equation and double feedforward neural network. In addition to river flows, the influence of sediment loads in the upstream river sections is investigated in this study. This chapter is structured in the following manner. Followed by Introduction, the details of estimation models for *SSL* are described in the section 4.2. The novel proposed model is identified and the process of implementation is presented. Section 4.3 gives the data information of two study areas and main results

and discussions are shown. Finally section 4.4 summarizes conclusions in this chapter.

4.2 Model development

This section proposes a hybrid double feedforward neural network (HDFNN) model for daily *SSL* estimation, by combining fuzzy pattern-recognition and continuity equation into a structure of double neural networks. The mechanism of the HDFNN model is displayed in details, derived from multi-layer feedforward neural network (MFNN), double parallel feedforward neural network (DPFNN) and hybrid feedforward neural network (HFNN) models. In the following, the procedures of the four models for the estimation of suspended sediment load are presented.

4.2.1 Multi-layer feedforward neural network model

The three-layer feedforward neural network consisting of the input, hidden and output layers, is the most widely used MFNN model. The topological structure of MFNN model is presented in Figure 4.1(a), in which only one node in the output layer is considered. Similar with the ANN model employed in the previous chapter, the nodes in the hidden and output layer are respectively computed from the following activation functions (Thirumalaiah and Deo, 1998):

$$t_i = f(\sum_{j=1}^{k} p_j w_{ji} + b_i) \tag{4.1}$$

$$y = F(\sum_{i=1}^{s} t_i \overline{w}_i + \overline{b}) \tag{4.2}$$

where p_j (j=1, 2, …, k) and t_i (i=1, 2, …, s) respectively represent nodes in the input and hidden layer, and y represents node in the output layer. The weight parameters are

denoted by w_{ji} and $\overline{w}_i$; b_i and $\overline{b}$ is the bias value. For traditional MFNN models, the activation function *f*(.) is usually a radial basis function or sigmoid function, and *F*(.) is a linear function. They reveal relation of nodes between two layers, although having no physical meanings. The MFNN model for *SSL* estimation has limitations attributed to the negligence of sediment properties.

4.2.2 Double parallel feedforward neural network model

As can be seen in Figure 4.1(a), DPFNN model is developed from MFNN model in which two networks connect each other in parallel with the same *k* input nodes. For the three-layer neural network of DPFNN, the nodes in the hidden layer $(t_1, t_2, ..., t_s)$ are computed by Eq. (4.1) and then connected to the output with weight parameters $\overline{w}_i$ in the same manner. Analogously for the two-layer neural network, the weight parameters directly from the input layer to the output layer are denoted by v_j (*j*=1, 2, …, *k*). The node in the output layer is obtained from the following equation (Zhong and Ding, 2005):

$$y = \sum_{i=1}^{s} t_i \overline{w}_i + \sum_{j=1}^{k} p_j v_j \tag{4.3}$$

The output is a summary of two parallel neural networks. For a given set of training samples (p^n, Y^n) supplied to the model, the error function is defined as:

$$E(W) = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (y^n - Y^n)^2} \tag{4.4}$$

where the vector *W* is a collection of all unknown parameters, and varies with the estimation model; y^n and Y^n are computed and desired output (*n*=1,2,…, *N*), respectively, and *N* is the number of training samples. The objective of network training, hence, is to find W_{opt} which satisfies that $E(W_{opt}) = min\, E(W)$. The vector *W* is updated with the

updated fitness value of $E(W)$ and is finally outputted if stopping criteria is satisfied. In the present chapter, differential evolution (DE) is employed as an optimization technique to find the minimum value of error function and the corresponding W_{opt}. The DE algorithm is a widely used population-based optimization method, which is favourable for searching parameters of non-differentiable and time-varying models (Li et al., 2013a; Rocca et al., 2011; Storn and Price, 1995).

4.2.3 Hybrid feedforward neural network model

The above two models are incapable of distinguishing the influences of different inputs, thus $\{p_1, p_2, ..., p_k\}$ is employed to represent any potential inputs for *SSL* estimation. In practice, some previous studies estimated sediment based on the river flow and sediment data at its own station (Afan et al., 2015; Aytek and Kisi, 2008), while others focused on the estimation of downstream sediment data by using data from both upstream and downstream stations (Kisi, 2004b; Partal and Cigizoglu, 2008). For the case of this study, river flows *Q* either at the upstream or downstream stations and *SSL* at upstream stations are involved as inputs. When fed with various inputs, the output *SSL* at the downstream station is obtained in different manners.

In this section, a hybrid feedforward neural network (HFNN) is developed with respect to river flow inputs $(Q_1^{in}, Q_2^{in}, ..., Q_k^{in})$. The conceptual activation function based on fuzzy pattern-recognition is introduced as follows (Qiu et al., 1998):

$$Q_i = \frac{1}{\sum_{l=1}^{C} \frac{\sum_{j=1}^{k} \left[w_{ji} \left(Q_j^{in} - M_i \right) \right]^2}{\sum_{j=1}^{k} \left[w_{ji} \left(Q_j^{in} - M_l \right) \right]^2}} \tag{4.5}$$

where Q_i (i=1, 2, …, s) are nodes in the hidden layer and Q_j^{in} (j=1, 2, …, k) are nodes in the input layer. Model vector $M = [M_i] = [M_l]$ is denoted the same with that in the previous chapter, which contains a number of patterns in the hidden layer and could fully cover the models ranging from "wet" to "dry" season. Meanwhile, the activation function from the hidden layer to the output layer is given as follows:

$$SSL^{(1)} = a_0 \times (\sum_{i=1}^{s} Q_i \overline{w}_i + \overline{b})^{b_0} \tag{4.6}$$

where $SSL^{(1)}$ represents node in the output layer; $\overline{w}_i$ and $\overline{b}$ stand for the weight parameters and bias in the output layer, respectively. The activation function in Eq. (4.6) expresses a power relationship between river flows and sediment loads, which is generally a functional relationship representing the SRC. Values of a_0 and b_0 for a specific river are to be optimized in the training process of neural network. The structure of HFNN model is depicted in the framework of Figure 4.1(b), where $SSL^{(1)}$ is considered as the final output with inputs $(Q_1^{in}, Q_2^{in}, ..., Q_k^{in})$. Accordingly, HFNN model examines the relationship of Q and SSL by considering the fuzzy property of sediment loads in an MFNN structure.

4.2.4 Hybrid double feedforward neural network model

In this section, a hybrid double feedforward neural network (HDFNN) is developed when sediment data at the upstream river stations are added as inputs. These sediment inputs directly work on the output in a two-layer neural network. In the representation of a river system, upstream stations are regarded as nodes in the input layer and downstream station as node in the output layer. Thus, mass conservation is satisfied over the river network by the following continuity equation (Li and Gu, 2003):

$$\frac{\partial SD^s}{\partial T} = \sum_{i=1}^{h} v_i Q_i^s - Q^s \tag{4.7}$$

where SD^s and Q^s are respectively the sediment deposition and sediment transport rate at the downstream station, and T is time. Meanwhile, Q_i^s is the sediment transport rate at each upstream station, wherein i (1, 2, …, h) refers to the index of each node in the input layer. The fraction of sediment from a node in the input layer entering into the node in the output layer is denoted by v_i. In the physical point of view, Eq. (4.7) implies that the rate of change of sediment deposition in the current river section is determined by the difference with the source sediment transport rate at the upstream river reaches, which reveals the sediment mass conservation over the entire river system. After discretization, the SD^s at time $T + \Delta T$ is determined by the following equation:

$$SD^s_{(T+\Delta T)} = SD^s_{(T)} + (\sum_{i=1}^{h} v_i Q^s_{i\ (T)} - Q^s_{(T)}) \times \Delta T \tag{4.8}$$

In this study, suspended sediment load refers to sediment discharge, thus the transport rate Q^s could be substituted by *SSL*, and $SSL_{i(T)}$ denotes sediment load at the upstream station. Accordingly, Eq. (4.8) is rewritten as follows:

$$SD^s_{(T+\Delta T)} = SD^s_{(T)} + (\sum_{i=1}^{h} v_i SSL_{i(T)} - SSL_{(T)}) \times \Delta T \tag{4.9}$$

Equation (4.9) in its simplified form is given by:

$$SD^s_{(T+\Delta T)} = \lambda_{(T)} \times (SD^s_{(T)} + P_{(T)}) \tag{4.10}$$

wherein $\lambda_{(T)} = 1 - \frac{SSL_{(T)} \times \Delta T}{SD^s_{(T)} + P_{(T)}}$ and $P_{(T)} = \sum_{i=1}^{h} v_i SSL_{i(T)} \times \Delta T$. Here λ is regarded as a recession coefficient, which is assumed to be independent of time (Yang et al., 1998). An initial value of sediment deposition SD_0^s is given in advance, and the value of SD^s at each time step is computed from Eq. (4.10). The *SSL* in the output layer is evaluated as a nonlinear function of sediment deposition (Li and Gu, 2003)

$$SSL^{(2)}_{(T+\Delta T)} = \frac{1}{1+\exp[-(SD^{s}_{(T+\Delta T)} + P_{(T+\Delta T)})]} \tag{4.11}$$

When introducing upstream input $SSL_{i(T)}$ (referring to SSL_i^{in} in the following) at time step *T* into the network, downstream sediment load $SSL^{(2)}_{(T+\Delta T)}$ is finally computed.

The HDFNN model employs two separate neural networks with influences of river flows and upstream sediment loads on downstream *SSL*, which is different from the DPFNN model using the same input variables in two parallel networks. This is tantamount to say that two neural networks with respect to $(Q_1^{in}, Q_2^{in}, ..., Q_k^{in})$ and $(SSL_1^{in}, SSL_2^{in}, ..., SSL_h^{in})$ are involved, as shown in Figure 4.1(b). The final output *SSL* is a summary of $SSL^{(1)}$ and $SSL^{(2)}$. Accordingly, HDFNN model is capable of addressing two separate inputs due to the double networks. Besides, the inclusion of fuzzy pattern-recognition and continuity equation in the neural networks enables consideration of fuzzy and time-varying feature of sediment loads.

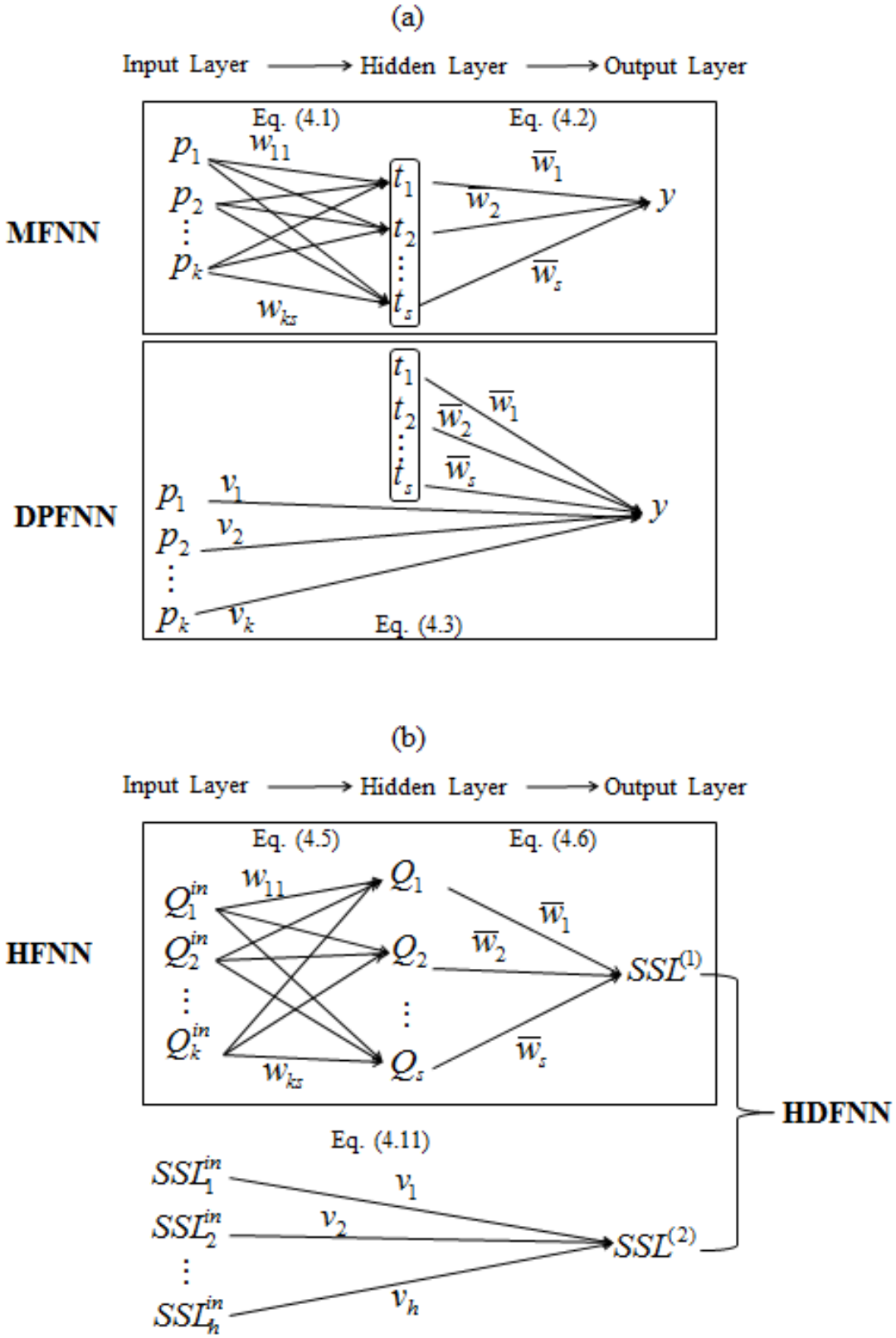

Figure 4.1 (a) Topological structure of MFNN and DPFNN models (b) topological structure of HFNN and HDFNN models

4.3 Suspended sediment estimation

4.3.1 Study area and data

Two watersheds in Montana of USA are considered as case studies. The first belongs to two stations on the Muddy Creek near Vaughn. The Muddy Creek is a tributary of the Missouri river, approximately 307 km long, and flows southeast through the Big Muddy Badlands. The drainage areas at these sites are 730.377 km^2 for the upstream station (station No. 06088300) and 813.256 km^2 for the downstream station (station No. 06088500), as shown in Table 4.1 and Figure 4.2. These two stations have been studied in several works (Browning et al., 2005; Kisi and Fedakar, 2014), which ensures the reliability of our collected data. The objective of this work is to estimate the suspended sediment loads at the downstream station (SSL_d) based on river flows at the upstream and downstream stations (Q_u and Q_d) and sediment loads at the upstream station (SSL_u).

The daily dataset is collated from US Geological Survey (USGS), covering a time period of four years from 1st January 1977 to 31st December 1980. The discharge and sediment data for the upstream and downstream stations are plotted in Figure 4.3. It can be seen that there is a highly nonlinear relationship between discharge and sediment data for both stations. The presence of outliers is detected as well, particularly for the sediment data. In the downstream dataset, only four values above or near 40000 tons/day are observed while the others are below 20000 tons/day. These outliers of data may create additional difficulties to the estimation models.

For the purpose of calibration and estimation, data for Years 1977 and 1978 are chosen in the training period, whilst those for Year 1980 serve the model testing. The remaining data for Year 1979 (around 25% of the whole data) are used for validation, which is an

indispensible process to avoid over-fitting. The statistical parameters of river discharge and sediment data for the two stations are summarized in Table 4.2, in which X_{mean}, X_{median}, S_X, X_{max} and X_{min} denote the mean, median, standard deviation, maximum and minimum, respectively. A noticeable difference between X_{mean} and X_{median} is detected for the sediment data, which provides supporting evidence for the existence of outliers. The high values of S_X indicate the complexity of the sediment data, and this may have a negative effect on the estimation performance. Moreover, the X_{min} value in the training set is higher than that in the corresponding testing set, for Q_u and SSL_u datasets. This may cause extrapolation difficulties in computing low sediment values.

Table 4.1 Stations' ID, name and location on the Muddy Creek near Vaughn in Montana

Station ID	Station name	Latitude	Longitude
06088300	Muddy Creek near Vaughn	47°37'30.08"N	111°38'07.46"W
06088500	Muddy Creek at Vaughn	47°33'40.56"N	111°32'30.37"W

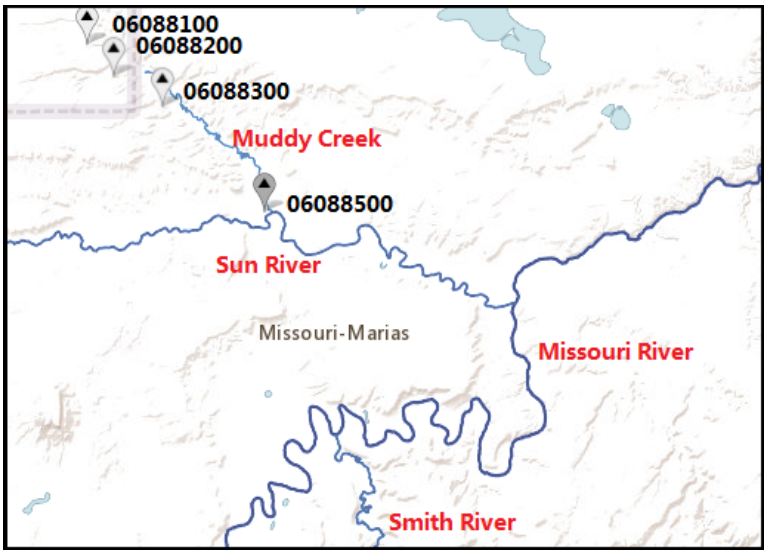

Figure 4.2 Locations of stations on the Muddy Creek near Vaughn in Montana, USA

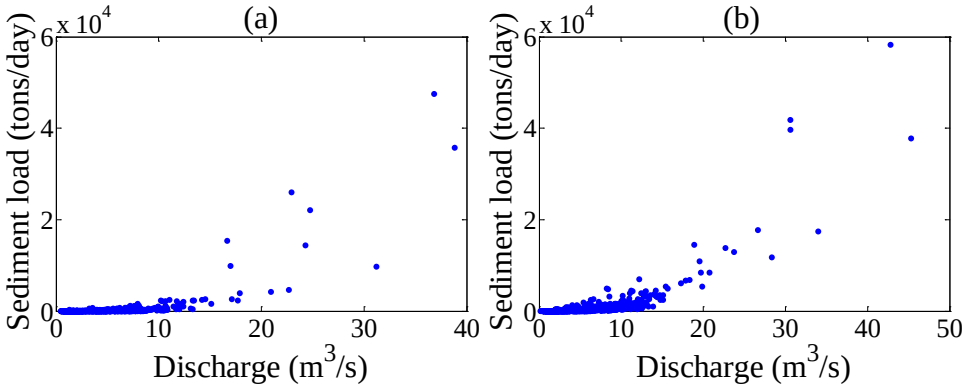

Figure 4.3 Scatter plots of (a) upstream and (b) downstream data between sediment load and discharge for the study case on the Muddy Creek

Table 4.2 Summary of statistical parameters for data at stations on the Muddy Creek

Station No.	Data type	Data period	X_{mean}	X_{median}	S_X	X_{max}	X_{min}
06088300 Data period: 1977/01-1980/12	Q_u (m^3/s)	Training	2.97	1.53	3.53	38.79	0.62
		Validation	3.57	1.81	3.57	31.15	0.71
		Testing	2.65	1.84	2.32	22.94	0.42
	SSL_u (tons/day)	Training	291.91	10.0	2450.3	47600	2.30
		Validation	175.84	13.0	666.67	9740	2.00
		Testing	175.55	13.5	1589.9	26100	1.40
06088500 Data period: 1977/01-1980/12	Q_d (m^3/s)	Training	3.79	1.93	4.33	45.31	0.14
		Validation	4.59	2.21	4.56	33.98	0.82
		Testing	3.64	2.07	3.34	30.58	0.45
	SSL_d (tons/day)	Training	610.28	25	3219.6	58300	0.96
		Validation	539.36	58	1413.7	17500	2.80
		Testing	384.04	30	2351.4	41900	1.70

The second study case is situated on the Tongue River in Montana, USA. The Tongue River is a tributary of the Yellowstone River, which rises in Wyoming in the Big Morn Mountains, flows northeast through northern Wyoming and southeastern Montana and empties into the Yellowstone River at Miles city. Table 4.3 and Figure 4.4 present the locations of the upstream station 06307830 and downstream station 06308500 on the Tongue River. The drainage areas for each station are 10046.6 km^2 and 13996.4 km^2, respectively. The collected data include discharge and suspended sediment load from both stations as well, and data period spans four years from 1st October 1977 to 30th September 1981. The scatter plots between sediment load and discharge data for the

upstream and downstream stations are demonstrated in Figure 4.5. The mapping of discharge and sediment data presents highly nonlinear feature, which could not be expressed by a simple power relationship. The sediment data has outlier values in particular, in which the maximum SSL_d is as high as 84400 tons/day while 98.63% of the data is below 10000 tons/day. The discharge and sediment data at the above two stations have been used in the study of Aytek and Kisi (2008).

Table 4.3 Stations' ID, name and location on the Tongue River in Montana

Station ID	Station name	Latitude	Longitude
06307830	Tongue River blow Brandenberg bridge near Ashland	45°50'25.39"N	106°13'24.36"W
06308500	Tongue River at Miles city	46°23'04.54"N	105°50'43.88"W

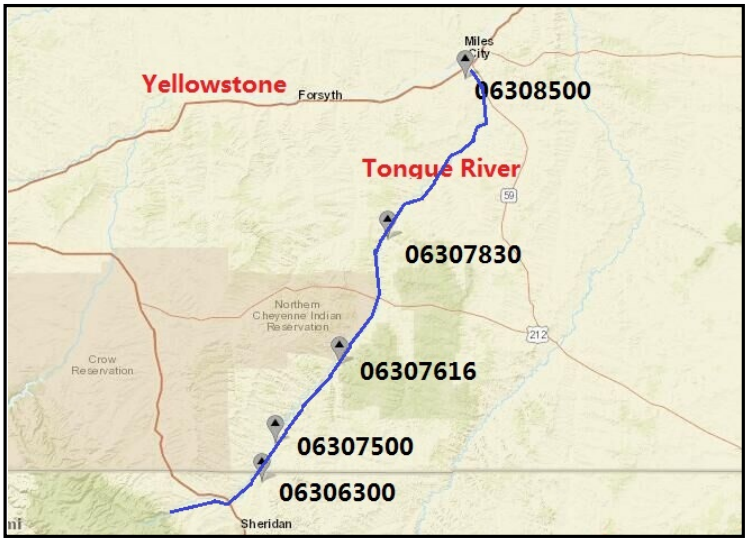

Figure 4.4 Locations of stations on the Tongue River in Montana, USA

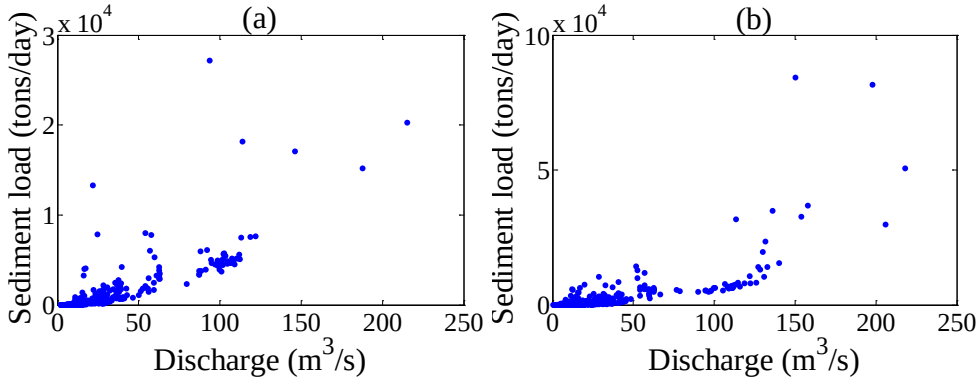

Figure 4.5 Scatter plots of (a) upstream and (b) downstream data between sediment load and discharge for the study case on the Tongue River

Similarly, the total data are partitioned into training, validation and testing sets. The data of the first two years are chosen for the training purpose, the first half of the remaining are employed to avoid over-fitting in the validation period, and the other half (i.e., data from 1st October 1980 to 30th September 1981) are chosen in the testing period. As seen from Table 4.4, the training set cannot fully include the validation and testing data for Q_u, Q_d and SSL_d data series. Besides, the values of S_X obtained from the discharge data in the training, validation and testing period are not approximate, when compared with the study case on the Muddy Creek. This reveals the imperfect data partition for the case on the Tongue River. It is intuitive that the lack of similar patterns between the training and testing datasets will give rise to poor performances in the testing period. The presence of outliers is detected by the salient differences between X_{mean} and X_{median} for the sediment data. For instance, the X_{mean} value reaches 1422.6 tons/day while the X_{median} is 86 tons/day in the training period for the SSL_d data. The high values of S_X also imply the widely-spread sediment data. In short, the sediment data to be estimated are irregular and ambiguous, and a model, which can map the highly nonlinear relationship between the downstream sediment load and the input variables, is in urgent need.

Table 4.4 Summary of statistical parameters for data at stations on the Tongue River

Station No.	Data type	Data period	X_{mean}	X_{median}	S_X	X_{max}	X_{min}
06307830 Data period: 1977/10-1981/09	Q_u (m^3/s)	Training	17.85	11.30	24.51	215.21	2.55
		Validation	9.53	8.38	5.02	30.58	1.84
		Testing	10.28	6.65	12.54	62.86	1.98
	SSL_u (tons/day)	Training	606.22	58	2088.6	27200	1.5
		Validation	120.13	43.5	253.98	1980	1.5
		Testing	234.88	26	834.75	8000	3
06308500 Data period: 1977/10-1981/09	Q_d (m^3/s)	Training	19.72	9.34	29.47	218.04	2.83
		Validation	7.53	6.33	4.54	28.32	1.90
		Testing	9.00	5.47	12.88	62.30	0.06
	SSL_d (tons/day)	Training	1422.6	86	5891.2	84400	4
		Validation	146.1	28	663.66	10600	3.4
		Testing	369.0	23	1235.1	7400	0.13

4.3.2 Performances of estimation models

4.3.2.1 Evaluation criteria and preliminary work

In order to undertake the comparison of performances by different models, two evaluation criteria are employed in current chapter, namely root mean square error (RMSE) and Nash-Sutcliffe efficiency coefficient (NSEC). Accuracy (ACC) is not an advisable measure for estimating models, since the sediment data present distinct outliers feature. Partitioning analysis is further conducted to provide explicit performances on different intervals of values, the details of which will be described in the following discussions.

The purpose of this study is to compute current downstream sediment load value $SSL_{d(t)}$ with single- and multi-step-ahead estimations. Since the discharges $Q_{u(t)}$ and $Q_{d(t)}$ for the current day have significant relations with the current sediment $SSL_{d(t)}$, the computation of sediment data belongs to "estimation" instead of "forecasting". For one-day-ahead estimation, $Q_{u(t)}$, $Q_{d(t)}$, $Q_{u(t\text{-}1)}$, $Q_{d(t\text{-}1)}$ and $SSL_{u(t\text{-}1)}$ are used to constitute the inputs. The MFNN and DPFNN models include the five input variables, while the HFNN model takes account of the four discharge inputs alone. The HDFNN model employs different inputs for multi-layer and two-layer: one encompasses discharge variables [$Q_{u(t)}$, $Q_{d(t)}$, $Q_{u(t\text{-}1)}$, $Q_{d(t\text{-}1)}$] and the other sediment variable $SSL_{u(t\text{-}1)}$. It should be noted that time step ΔT in the HDFNN model represents lead time, which is the same as that in the previous chapter. Thus ΔT =1 day indicates that the downstream sediment $SSL_{d(t)}$ at time t is computed from upstream sediment $SSL_{u(t\text{-}1)}$ at time t-1. Likewise, the inputs for the two- and three-day-ahead estimations are [$Q_{u(t)}$, $Q_{d(t)}$, $Q_{u(t\text{-}2)}$, $Q_{d(t\text{-}2)}$, $SSL_{u(t\text{-}2)}$] and [$Q_{u(t)}$, $Q_{d(t)}$, $Q_{u(t\text{-}3)}$, $Q_{d(t\text{-}3)}$, $SSL_{u(t\text{-}3)}$], respectively. In addition, the datasets of discharge and sediment are scaled linearly within the range of [0.1, 0.9] according to Eq. (3.10) to avoid larger

data dominating smaller data.

Primarily, it is an imperative task to determine the number of nodes in the hidden layer for a neural network. It is generally conducted by comparing the fitness value in the training period under trial tests. In particular, Figures 4.6 and 4.7 plot the fitness values versus the number of nodes in the hidden layer by the HFNN and HDFNN models for the case on the Muddy Creek. It is noted that these two models introduce fuzzy pattern-recognition into the hidden layer, thus the number of nodes in the hidden layer also means the number of elements in the model vector *M*. For the HFNN model, the value C=5 and $M = (1.0, 0.75, 0.50, 0.25, 0)$ are obtained with smallest fitness value. The model vector $M = (1.0, \frac{5}{6}, \frac{4}{6}, \frac{3}{6}, \frac{2}{6}, \frac{1}{6}, 0)$ with C=7 is large enough to perform the nonlinear property for the HDFNN model, as indicated by Figures 4.7. The same method is used for the case on the Tongue River, as demonstrated in Figures 4.8 and 4.9 for the HFNN and HDFNN models respectively. The minimum fitness takes place at C=8 for both of the models. That is, the nonlinear and fuzzy properties of flows can be fully acquired with eight modes from totally wet to dry time for the study case. It seems that the time series for the case on the Tongue River is more complicated and uncertain than that on the Muddy Creek, thus the estimations are performed in the context of more patterns with larger C values. The numbers of nodes in the hidden layer in the MFNN and DPFNN models are determined in the same manner.

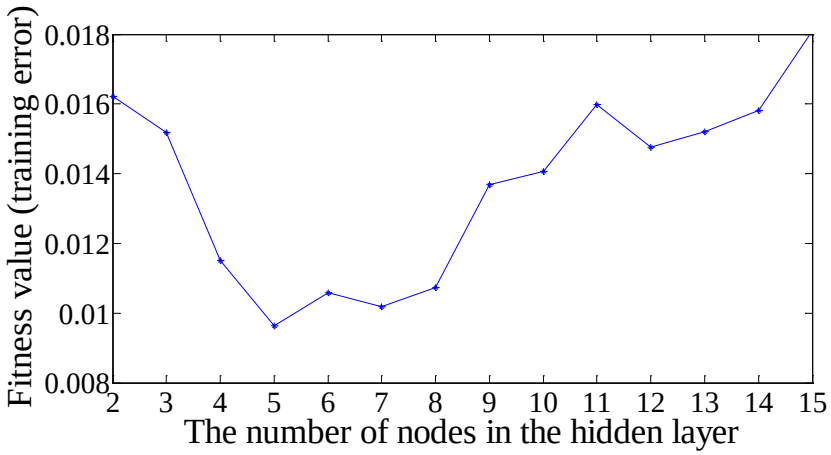

Figure 4.6 Fitness values with different numbers of nodes in the hidden layer by the HFNN model for the study case on the Muddy Creek

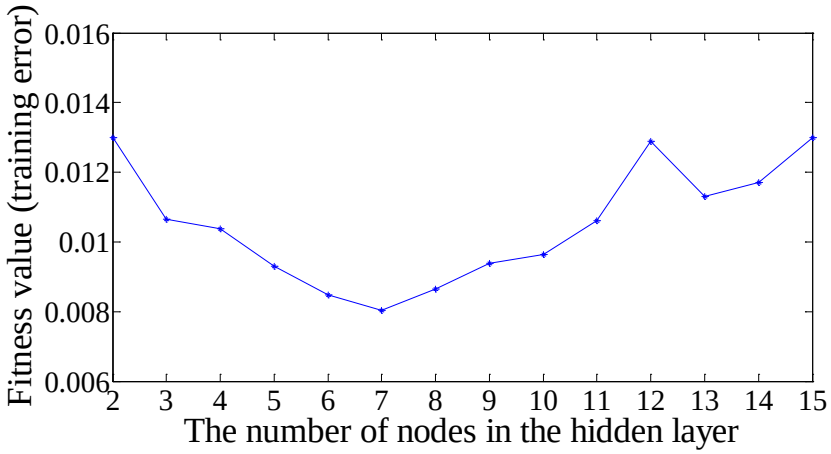

Figure 4.7 Fitness values with different numbers of nodes in the hidden layer by the HDFNN model for the study case on the Muddy Creek

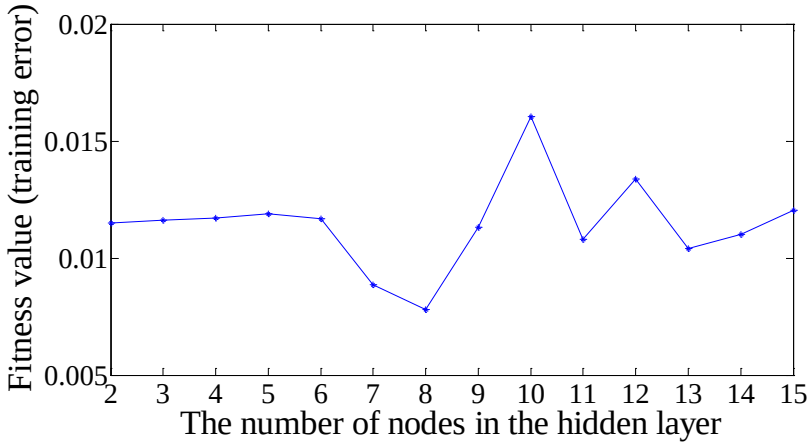

Figure 4.8 Fitness values with different numbers of nodes in the hidden layer by the HFNN model for the study case on the Tongue River

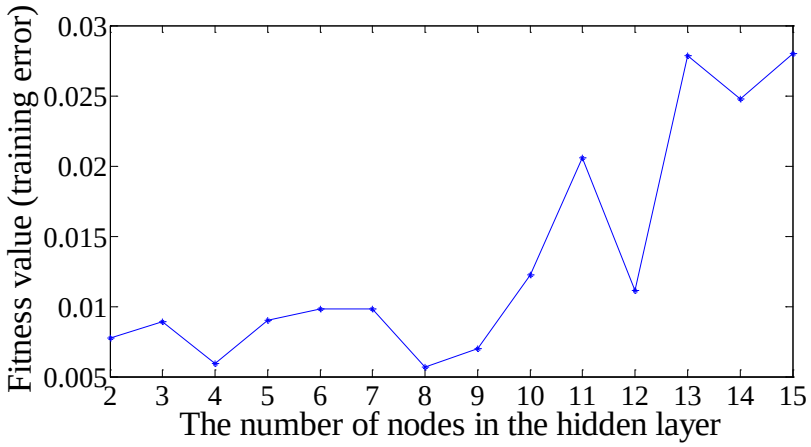

Figure 4.9 Fitness values with different numbers of nodes in the hidden layer by the HDFNN model for the study case on the Tongue River

4.3.2.2 Estimation results for study case 1

The performances of one-, two- and three-day-ahead estimations by the four models in terms of RMSE and NSEC are summarized in Table 4.5. The configurations of neural network for each model are provided as well. Taking the one-day-ahead case as an example, (5,4,1) for MFNN model implies that there are 5 nodes in the input layer, 4 nodes in the hidden layer and 1 node in the output layer. For the DPFNN model, the structures for multi-layer and two-layer are respectively (5,5,1) and (5,1) with same inputs. HFNN model only considers the discharge inputs, thus there are only four input nodes. The HDFNN model has different inputs for multi-layer and two-layer, the structures of which are (4,7,1) with seven nodes in the hidden layer and (1,1) with one node in the input layer, respectively.

For the one-day-ahead estimation with input combination [$Q_{u(t-1)}$, $Q_{u(t)}$, $Q_{d(t)}$, $Q_{d(t-1)}$, $SSL_{u(t-1)}$], MFNN model has a fairly high RMSE value (i.e., 815.4227 tons/day), and shows inferior results due to its total “black-box” operation. DPFNN model demonstrates slightly better generalization and estimation capability than MFNN model, as indicated by the more approximate RMSE values in the training and testing periods.

HFNN model is found to be superior to MFNN model as well, with an increase of 2.83% and 3.68% in NSEC value for the training and testing period respectively. The RMSE value of HDFNN model is respectively 55.50%, 55.43% and 47.95% lower than that of MFNN, DPFNN and HFNN model in the testing period. Meanwhile, the NSEC value attained by the HDFNN model is 0.9762, which is superbly high to reveal the ability of HDFNN model for sediment estimation.

The performances of each model gradually deteriorate with the increase of lead time, as observed from Table 4.5. The two-day-ahead estimations yield inferior performances than the one-day-ahead cases in general, yet, still attain desirable results by the HDFNN model with a high NSEC value in the testing period (i.e., 0.9733) herein. The capacity of mapping both nonlinear and linear relationship with the double parallel networks by the DPFNN model renders smaller RMSE and higher NSEC values than the MFNN model in the testing period. The HFNN model brings about more improvement of performances, particularly in the training period. Although the influence of upstream sediment loads is not included, the physical activation function provides information about different patterns and considerably improves the generalization ability. HDFNN model draws the advantages of both DPFNN and HFNN models, hence, gives the best performances amongst the four models. When compared with MFNN model, there is a 61.31% and 55.89% reduction in RMSE value for the training and testing stages, respectively. The superiority of HDFNN over DPFNN and HFNN models is obvious, particularly for the NSEC values in the testing period. Similarly for the three-day-ahead estimations, the HDFNN model yields the lowest RMSE and highest NSEC values. The reduction of RMSE values is respectively 44.80%, 43.28% and 42.50% in comparison with that of MFNN, DPFNN and HFNN models in the testing period. It seems that the superiority degree of the HDFNN model does not significantly mitigate with the

increase of lead time. As a consequence, the HDFNN is an effective and alternative model for downstream sediment loads with single- and multi-step-ahead estimations.

Table 4.5 Performances by various models for the study case on the Muddy Creek

Input	Model	Configuration	Training		Testing	
			RMSE (tons/day)	NSEC	RMSE (tons/day)	NSEC
One-day-ahead	MFNN	(5,4,1)	665.8922	0.9572	815.4227	0.8797
	DPFNN	(5,5,1)+(5,1)	692.0350	0.9537	814.1384	0.8801
	HFNN	(4,5,1)	402.7692	0.9843	697.1047	0.9121
	HDFNN	(4,7,1)+(1,1)	**284.9112**	**0.9922**	**362.8466**	**0.9762**
Two-day-ahead	MFNN	(5,4,1)	813.1665	0.9361	872.9676	0.8625
	DPFNN	(5,5,1)+(5,1)	819.9004	0.9351	842.3295	0.8720
	HFNN	(4,5,1)	404.6351	0.9842	787.5140	0.8881
	HDFNN	(4,7,1)+(1,1)	**314.6514**	**0.9904**	**385.0416**	**0.9733**
Three-day-ahead	MFNN	(5,4,1)	897.9432	0.9221	914.0537	0.8497
	DPFNN	(5,5,1)+(5,1)	814.9446	0.9358	889.5487	0.8573
	HFNN	(4,5,1)	481.0746	0.9776	877.5841	0.8575
	HDFNN	(4,7,1)+(1,1)	**367.5171**	**0.9869**	**504.5875**	**0.9542**

The time series of observed and computed *SSL* as well as the scatter plots by various models for the one-day-ahead case are respectively demonstrated in the left- and right-hand side of Figure 4.10. It is observed that the HDFNN model perfectly fits the peak sediment load while the other three models underestimate the corresponding one. The estimations of high values including the peak one by various models are presented in Table 4.6, since they are quite scattered and inconvenient to be observed in figures. It is noticed that the HDFNN model reproduces 41279.4 tons/day for the peak value, which is almost equal to the observed one. All the four models underestimate the second highest value of 14600 tons/day, and generate unpromising results for the observations of 4550 tons/day and 4040 tons/day. This concretely displays the difficulty in capturing high values for the Muddy Creek datasets. The data applied to scatter plots in Figure 4.10 are below 500 tons/day, which take up around 83% of all data and are used to present the performances of relatively medium and low values. It is observed that HDFNN model estimates are less scattered in relation to the other three models. MFNN

and HFNN over-estimate most of the values, whilst DPFNN could not model the observations lower than 50 tons/day. It can be concluded that the HDFNN model is more adequate than the others for *SSL* estimation.

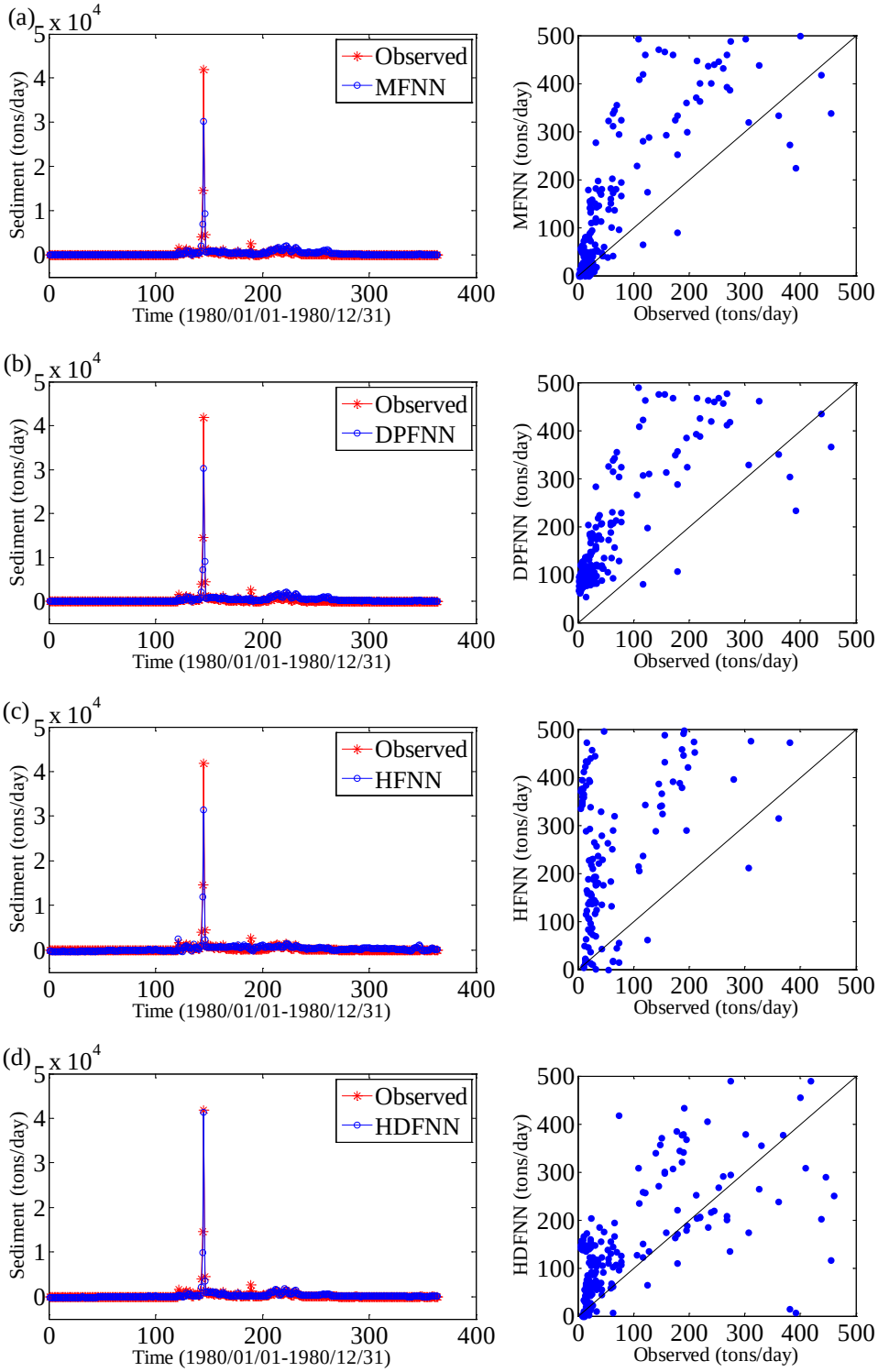

Figure 4.10 The observed and estimated suspended sediments by (a) MFNN (b) DPFNN (c) HFNN (d) HDFNN models for the study case on the Muddy Creek (one-day-ahead)

Table 4.6 High values by various models for the study case on the Muddy Creek

Input	model	High value1 (tons/day)	High value2 (tons/day)	High value3 (tons/day)	High value4 (tons/day)
	Observed	41900	14600	4550	4040
One-day-ahead	MFNN	30299.8	7052.1	9368.0	2064.7
	DPFNN	30221.2	7201.8	9098.8	2038.6
	HFNN	31364.1	**11858.6**	2238.2	1016.9
	HDFNN	**41279.4**	9950.6	**3510.8**	**2066.2**
Two-day-ahead	MFNN	27668.0	8741.2	2072.6	1194.3
	DPFNN	28236.8	8794.1	1950.1	**1882.5**
	HFNN	29174.1	**11273.3**	2795.3	973.0
	HDFNN	**40553.5**	9606.2	**3453.9**	1282.9
Three-day-ahead	MFNN	28086.9	8736.1	2612.8	**2529.0**
	DPFNN	27342.0	8711.0	2081.5	2005.7
	HFNN	27056.8	**10663.1**	**4182.7**	875.6
	HDFNN	**40314.4**	8637.3	2871.1	2307.1

It is worth noting that the final output of HDFNN model is a summary of $SSL^{(1)}$ and $SSL^{(2)}$, which are respectively computed from discharge and sediment inputs. Figure 4.11 shows the total output *SSL* of the middle data series for convenience of visual inspection, and its separate parts of result $SSL^{(1)}$ and $SSL^{(2)}$. The basic shape of three series is the same, in which $SSL^{(1)}$ makes the main contribution to the total sediment. Thus the validation of the mechanism of HDFNN model is attested, in which the two parts of inputs solely and effectively contribute the final output without contradiction.

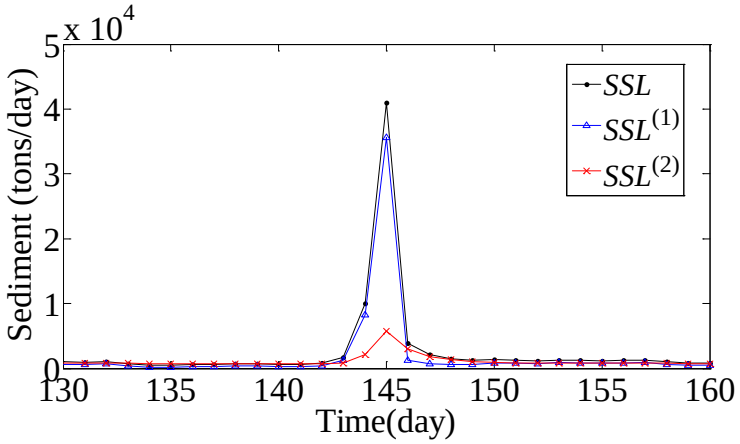

Figure 4.11 The separate parts of estimation result by the HDFNN model for the study case on the Muddy Creek (one-day-ahead)

The statistics RMSE and NSEC scale the mean squared error of estimation models, therefore particularly reflect the performance on high values. Thus the above discussions on evaluation criteria and plots of estimated data could not provide explicit performances on different intervals of values. To address this problem, partitioning analysis is undertaken with regard to the performances of four models in this study, by finding threshold values of dataset and partitioning the data into several intervals (Goyal 2014). Threshold values are determined based on the observed *SSL* data in the testing period. Median and mean of the dataset are considered as two threshold values. Values lower than median (30 tons/day) are regarded as a “low load”; values higher than mean (384.04 tons/day) as a “high load”; values higher than median and lower than mean as a “medium load”. The RMSE statistics by the four models with respect to low, medium and high loads are illustrated in Table 4.7. For the ‘low load’, HFNN model performs worse than its counterparts with a largest RMSE value due to the irrespective of upstream sediment data. HDFNN model is completely adequate in estimating low *SSL* values. Four models are capable of reproducing the ‘medium load’ with comparable performances, in which HDFNN model achieves the best result. The RMSE obtained by HDFNN model is much smaller than the other three models in ‘high load’. In overall, the performances of HDFNN model on low, medium and high loads are consistently excellent, which corroborates the use of this approach in *SSL* estimation.

As can be seen in Table 4.6, the HDFNN model still generates the best peak value respectively for the two- and three-day-ahead estimations. The other high values estimated by the four models are not matched with the observations, and even worse when compared with the one-day-ahead case. When examining the RMSE statistics of different loads in Table 4.7, DPFNN model obtains the smallest RMSE values in low loads. The HFNN model could not fit the low and medium loads in particular, which

may be attributed to the absence of sediment input. The influence of upstream sediment on the corresponding downstream sediment lies on the storage capacity of the reservoir, thus the process is slow and accumulated. On the contrary, the river flows wash the sediment loads, and therefore, the contribution of discharges on sediment may mainly concentrate on the high values. That is, when the sediment input is not included, the estimation model tends to be improper in computing low and medium loads. The HDFNN model obtains ultra-low RMSE values in medium and high loads for the two-day-ahead estimation. And for the three-day-ahead case, it fits the high loads best although yields a higher RMSE value in medium loads than the DPFNN model. To conclude, the HDFNN model shows splendid performances on medium and high loads.

Table 4.7 RMSE statistics of different loads by various models for the study case on the Muddy Creek

Input	model	Low load RMSE (tons/day)	Medium load RMSE (tons/day)	High load RMSE (tons/day)
One-day-ahead	MFNN	**43.5420**	296.1510	1808.1
	DPFNN	89.3811	304.0241	1798.8
	HFNN	243.7477	375.7581	1456.2
	HDFNN	112.2258	**161.2308**	**776.16**
Two-day-ahead	MFNN	23.9308	290.4892	1940.4
	DPFNN	**20.6136**	260.1144	1876.9
	HFNN	240.2565	378.4800	1673.9
	HDFNN	96.9564	**177.7255**	**827.81**
Three-day-ahead	MFNN	202.2498	468.9671	1954.4
	DPFNN	**22.4852**	**305.3180**	1975.5
	HFNN	255.6576	360.1935	1916.5
	HDFNN	60.0741	348.1393	**1050.9**

Figure 4.12 further demonstrates the scatter plots by various models for the two-day-ahead case. The data applied to the figures are below 2000 tons/day, since the performances of high values have been presented in Table 4.6. It is noticed that the HDFNN model outperforms the others with the evidence of intensively distributed dots along the ideal line. Most values below 500 tons/day are over-estimated by the HFNN model, which exhibits the mismatching with the low and medium loads. However, the

HFNN model obtains several promising dots between 500 and 1500 tons/day, which are obviously better than the other models. Likewise, Figures 4.13 present the model performances at three-day-ahead estimation. Compared with one- and two-day-ahead cases, the improvement by HDFNN model is not that significant. Visibly, the distribution of dots computed by HFNN model is different from the others, which may be attributed to the irrespective of sediment input. The better performances of HDFNN model rely on the excellent matching of high values, thus its superiority over the DPFNN model is not exhibited in the figure. It is rational that the computation of high values tends to be more difficult when performing more step-ahead estimations. The HDFNN model solves this problem by introducing fuzzy pattern-recognition into the neural network, which could effectively classify the discharge inputs and simulate the mode of high values. Thus the HDFNN is an appropriate model especially in fitting high values for multi-step-ahead estimations.

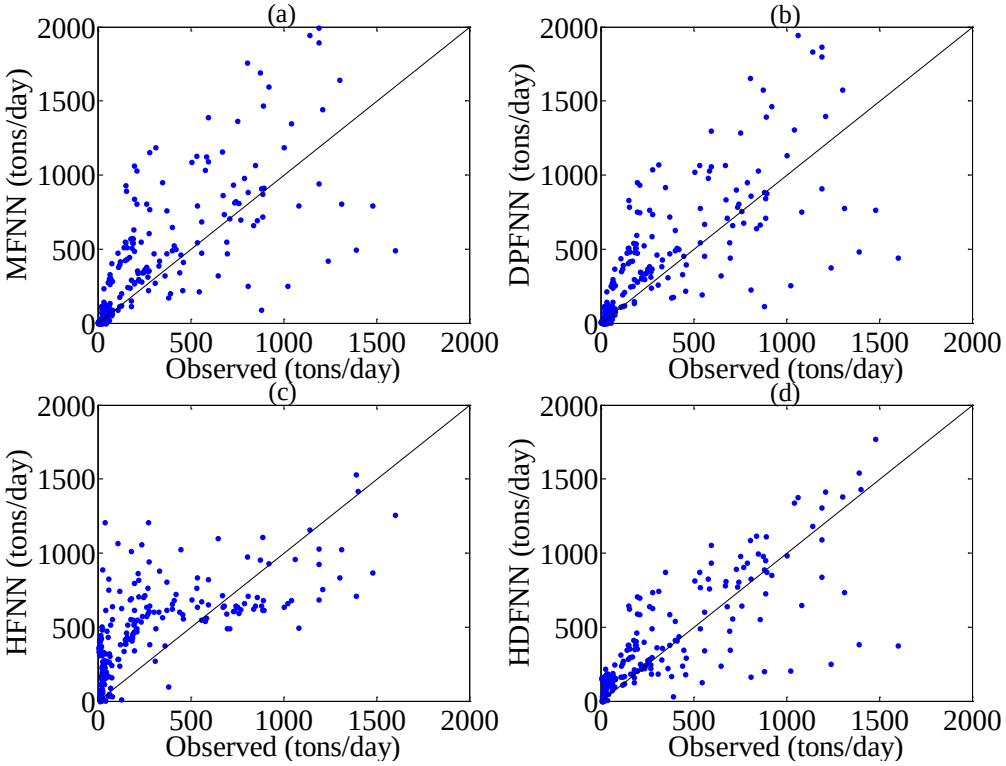

Figure 4.12 The observed and estimated suspended sediments by (a) MFNN (b) DPFNN (c) HFNN (d) HDFNN models for the study case on the Muddy Creek (two-day-ahead)

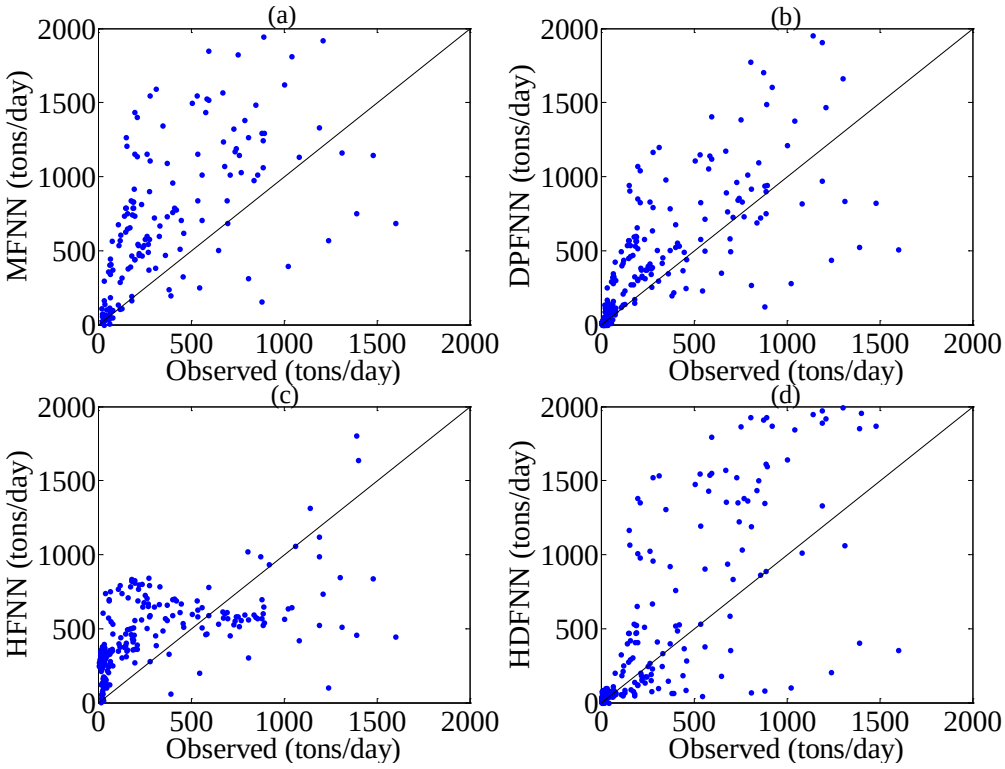

Figure 4.13 The observed and estimated suspended sediments by (a) MFNN (b) DPFNN (c) HFNN (d) HDFNN models for the study case on the Muddy Creek (three-day-ahead)

It should be noted that the HFNN and HDFNN models take physical activation functions into account. Several parameters to be optimized are meaningful related with the study basin and worthy discussing. In HFNN model, a_0 and b_0 denote parameters in the power relationship between river flows and sediment loads, as shown in Eq. (4.6). In addition to the parameters a_0 and b_0, recession coefficient λ is a special one for the HDFNN model, representing storage capacity of the assumed reservoir. The values of these parameters are presented in Table 4.8. The values of parameters from one- and two-day-ahead estimations appear to be quite close to each other, which validates the availability of power relationship and storage reservoir assumption. The values optimized by the three-day-ahead case are not appropriate to express the characteristics of the study basin, since the corresponding estimation performances are relatively poor.

Table 4.8 Special parameters of HFNN and HDFNN models for the study case on the Muddy Creek

Input	HFNN		HDFNN		
	a_0	b_0	a_0	b_0	λ
One-day-ahead	0.2809	7.6875	0.0547	5.2456	0.6504
Two-day-ahead	0.2425	7.9936	0.0474	5.1696	0.6011
Three-day-ahead	0.4053	3.1528	0.0900	4.8128	0.8373

4.3.2.3 Estimation results for study case 2

When applying the four models to the study case on the Tongue River, the estimations may be more complicated since the drainage areas are much larger when compared with that of the Muddy Creek. Table 4.9 presents the performances of one-, two- and three-day-ahead estimations in terms of RMSE and NSEC as well as the configurations of each network for the study case on the Tongue River. The best result is predictably achieved by the HDFNN model for the one-day-ahead case, with lowest RMSE values both in the training and testing period. The corresponding NSEC value in the testing stage is not as high as that of the case on the Muddy Creek, however, a value of 0.8985 is appropriate for the estimation. There is a reduction of 23.56%, 17.13%, 13.34% in RMSE value of HDFNN model in comparison with the MFNN, DPFNN and HFNN models respectively in the testing period. The improvement by the proposed model is not as distinct as the first study case. The two-day-ahead estimations attain more approximated results by the four models in the testing period. The HFNN model has extraordinarily high RMSE values in the training stage, which does not agree with its performance in the testing stage. The probable reason is the extreme inconsistence between training and testing datasets, which does not allow the model to be appropriately trained. The MFNN model obtains an unacceptably low NSEC value (i.e., 0.6812) and yields visibly worse results among the four models for the three-day-ahead case, revealing its unavailability in multi-step-ahead estimations. The ability of HDFNN model does not apparently decrease for the two- and three-day-ahead cases, however.

Instead, the HDFNN model remarkably outperforms the others on multi-step-ahead estimations. For instance, there is an increase of 27.55%, 8.72% and 3.77% in NESC value by the HDFNN model when compared with MFNN, DPFNN and HFNN models respectively, for the three-day-ahead case. Thus the suitability of the HDFNN model in multi-step-ahead estimations is confirmed. In all, the HDFNN model consistently performs excellent generalization and estimation abilities in the context of Tongue River case, despite of data complexity.

Table 4.9 Performances by various models for the study case on the Tongue River

Input	Model	Configuration	Training		Testing	
			RMSE (tons/day)	NSEC	RMSE (tons/day)	NSEC
One-day-ahead	MFNN	(5,5,1)	1092.3099	0.9655	514.8608	0.8262
	DPFNN	(5,5,1)+(5,1)	1041.1448	0.9687	474.9033	0.8521
	HFNN	(4,8,1)	1013.6458	0.9703	454.1077	0.8648
	HDFNN	(4,8,1)+(1,1)	**972.9268**	**0.9727**	**393.5422**	**0.8985**
Two-day-ahead	MFNN	(5,5,1)	1088.9087	0.9657	599.5435	0.7649
	DPFNN	(5,5,1)+(5,1)	1073.7914	0.9667	536.1495	0.8120
	HFNN	(4,8,1)	1451.7979	0.9391	481.4851	0.8484
	HDFNN	(4,8,1)+ (1,1)	**993.4844**	**0.9715**	**429.0833**	**0.8796**
Three-day-ahead	MFNN	(5,5,1)	1214.6259	0.9574	700.2113	0.6812
	DPFNN	(5,5,1)+ (5,1)	951.6755	0.9738	554.6614	0.7993
	HFNN	(4,8,1)	969.3895	0.9728	499.4645	0.8373
	HDFNN	(4,8,1)+ (1,1)	**951.6636**	**0.9738**	**448.3174**	**0.8689**

The time series of observed and computed *SSL* and the scatter plots by various models for the one-day-ahead case are demonstrated in Figure 4.14. The left-hand side of the figures mainly presents the performances of two valleys in the time series while the right part presents results below 500 tons/day. It is found that the first three models can not fit the first small valley data while the HDFNN model obtains an advanced valley with approximate peak value. The MFNN model fairly underestimates the second valley, wherein reproducing a peak value of 6177.5 tons/day when the corresponding observation is 7400 tons/day. The DPFNN and HFNN models can basically match the second valley data, however, not as perfectly as the HDFNN model. The plots with data

below 500 tons/day also reveal excellent performance of the HDFNN model, with the evidence of less scattered. The results of HDFNN model can be further presented in Figure 4.15, with the total and separate parts of the output. Observations reveal that the $SSL^{(2)}$ mainly makes contribution to the valley data, and virtually attains equivalent values for the small loads. This conforms to the influence of upstream sediment input on the HDFNN model, that the reservoir storage plays a consistent and stable role when the load is small and can acts efficiently with varying loads. Therein, the $SSL^{(2)}$ works as a supplement of the total sediment output, nevertheless, an dispensable one.

At the meantime, partitioning analysis is undertaken to show the performances on different loads. Values lower than 23 tons/day are assigned to “low load”; higher than 369 tons/day to “high load”; the remaining parts to “medium load” for the present study case. As can be seen in Table 4.10, the HDFNN model yields smallest RMSE values in medium and high loads for the one-day-ahead case. The best RMSE for the low loads is obtained by the HFNN model, however. Similar results are observed in the two- and three-day-ahead cases. This is not accordant with the case on the Muddy Creek, in which the HFNN model yields worse RMSE results in low loads. As mentioned previously, the absence of sediment input in the HFNN model leads to inadequate computation of low loads, thus to high RMSE results. But for the case on the Tongue River, the influence of sediment input on low values weakens, thus the HDFNN model does not outperforms the HFNN model. Conversely, the best performance of HFNN model in low loads lies in the facts that the inputs are classified into different patterns with large enough model vector. Besides, the HDFNN model merely yields the best RMSE value in high loads for the three-day-ahead estimation, which is in keeping with the previous study case. The possible reason is that when the lead time is larger, the effect of reservoir storage is reinforced particularly on the high loads of the HDFNN

model. Thus the HDFNN model can show advantage on the computation of high loads. Another phenomenon can be observed in Table 4.10 is that the HDFNN model reproduces comparable peak value for the one-, two- and three-day-ahead cases, which implies its stability in the estimation.

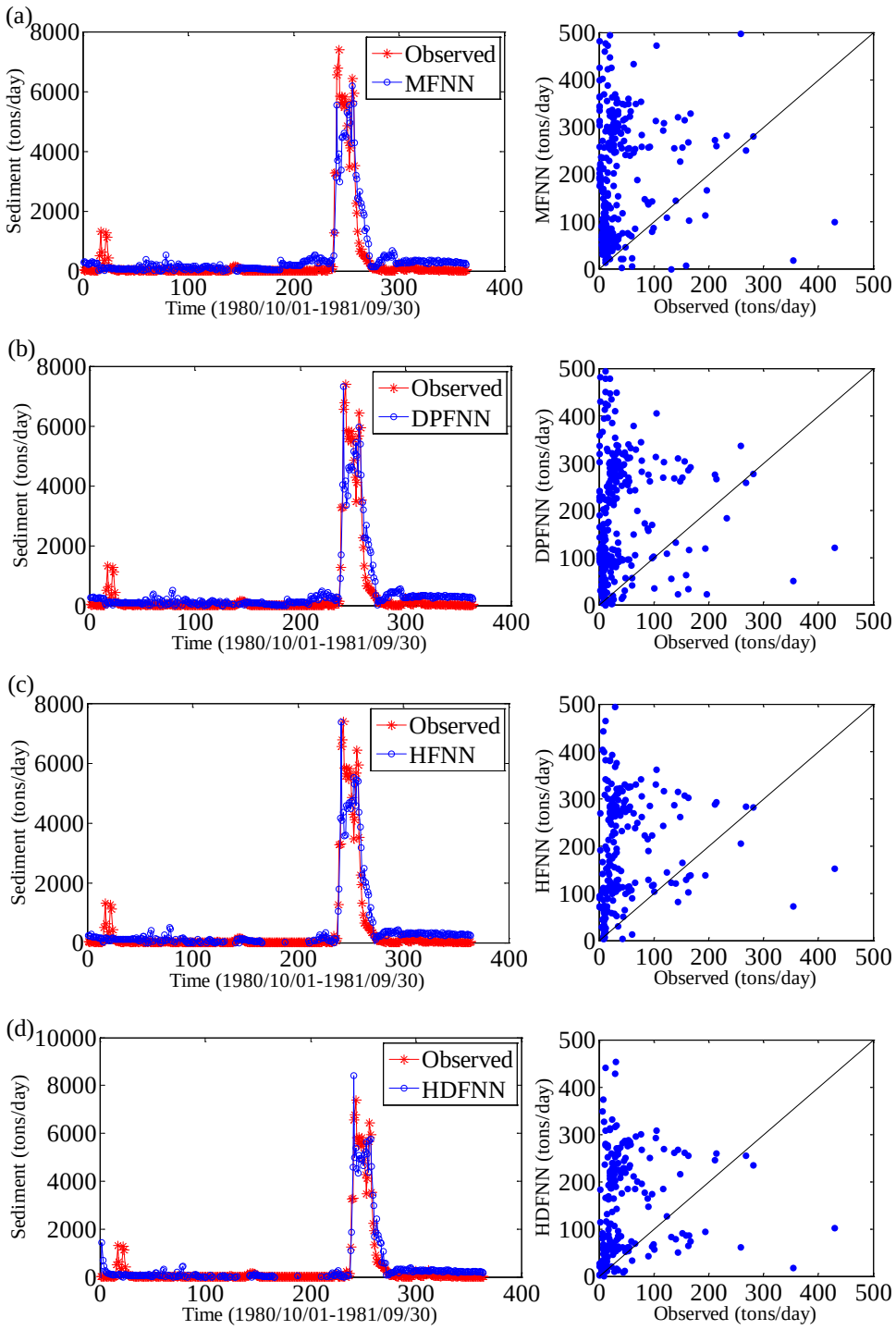

Figure 4.14 The observed and estimated suspended sediments by (a) MFNN (b) DPFNN (c) HFNN (d) HDFNN models for the study case on the Tongue River (one-day-ahead)

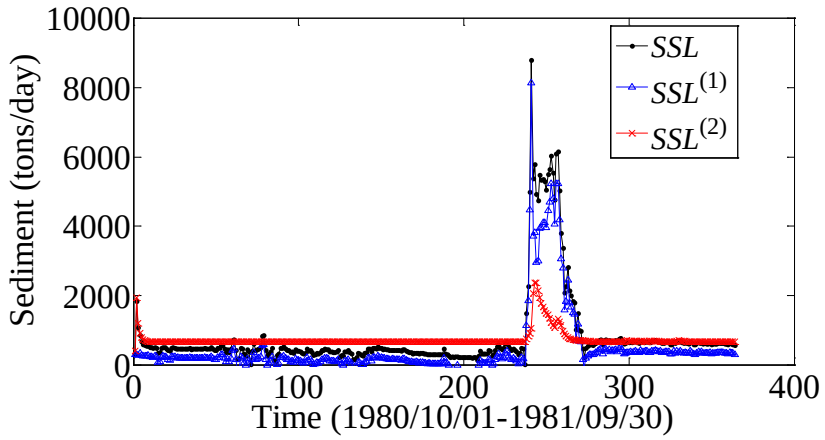

Figure 4.15 The separate parts of estimation result by the HDFNN model for the study case on the Tongue River (one-day-ahead)

Table 4.10 RMSE statistics of different loads and peak value by various models for the study case on the Tongue River

Input	model	Low load RMSE (tons/day)	Medium load RMSE (tons/day)	High load RMSE (tons/day)	Peak value (tons/day)
One-day-ahead	MFNN	222.2182	216.7987	1457.5	6177.5
	DPFNN	190.1340	202.5173	1353.6	7325.6
	HFNN	**160.1106**	194.8698	1307.5	**7364.7**
	HDFNN	174.0497	**178.9218**	**1103.3**	8397.0
Two-day-ahead	MFNN	302.8932	258.4876	1655.6	12831.0
	DPFNN	239.0731	237.1790	1503.5	12032.3
	HFNN	**216.2265**	254.2086	1322.4	**7562.3**
	HDFNN	268.8467	**228.8932**	**1101.7**	9430.1
Three-day-ahead	MFNN	488.5232	353.1156	1748.1	12126.8
	DPFNN	272.5450	217.2885	1548.3	11154.1
	HFNN	**231.2159**	**207.8675**	1399.4	10396.6
	HDFNN	252.5055	236.4333	**1183.2**	**9606.0**

Furthermore, Figures 4.16 and 4.17 respectively present the scatter plots for the two- and three-day-ahead cases. The left-hand side of the figures shows results under 500 tons/day, which include low and medium loads in the partition analysis. The four models generally overestimate the observations. The advantages of HFNN model over the HDFNN model are not exhibited in the plots since the RMSE values in low and medium loads are slightly better than the HDFNN model, as shown in Table 4.10. The right parts of the figures show the output between 2000 and 14000 tons/day. It is observed that the MFNN and DPFNN models overestimate the peak value markedly,

and the dots computed by the HDFNN models distribute closer to the ideal lines.

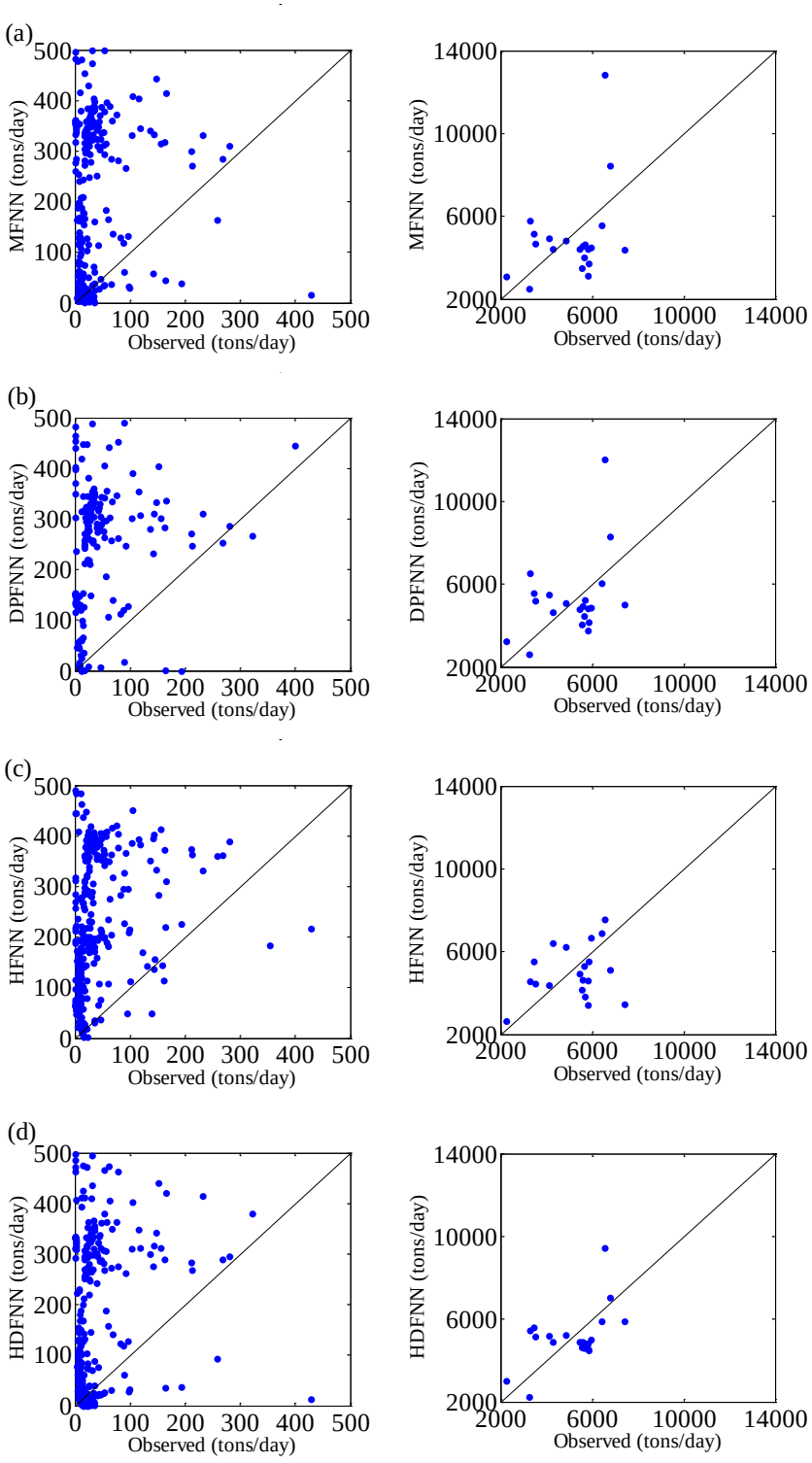

Figure 4.16 The observed and estimated suspended sediments by (a) MFNN (b) DPFNN (c) HFNN (d) HDFNN models for the study case on the Tongue River (two-day-ahead)

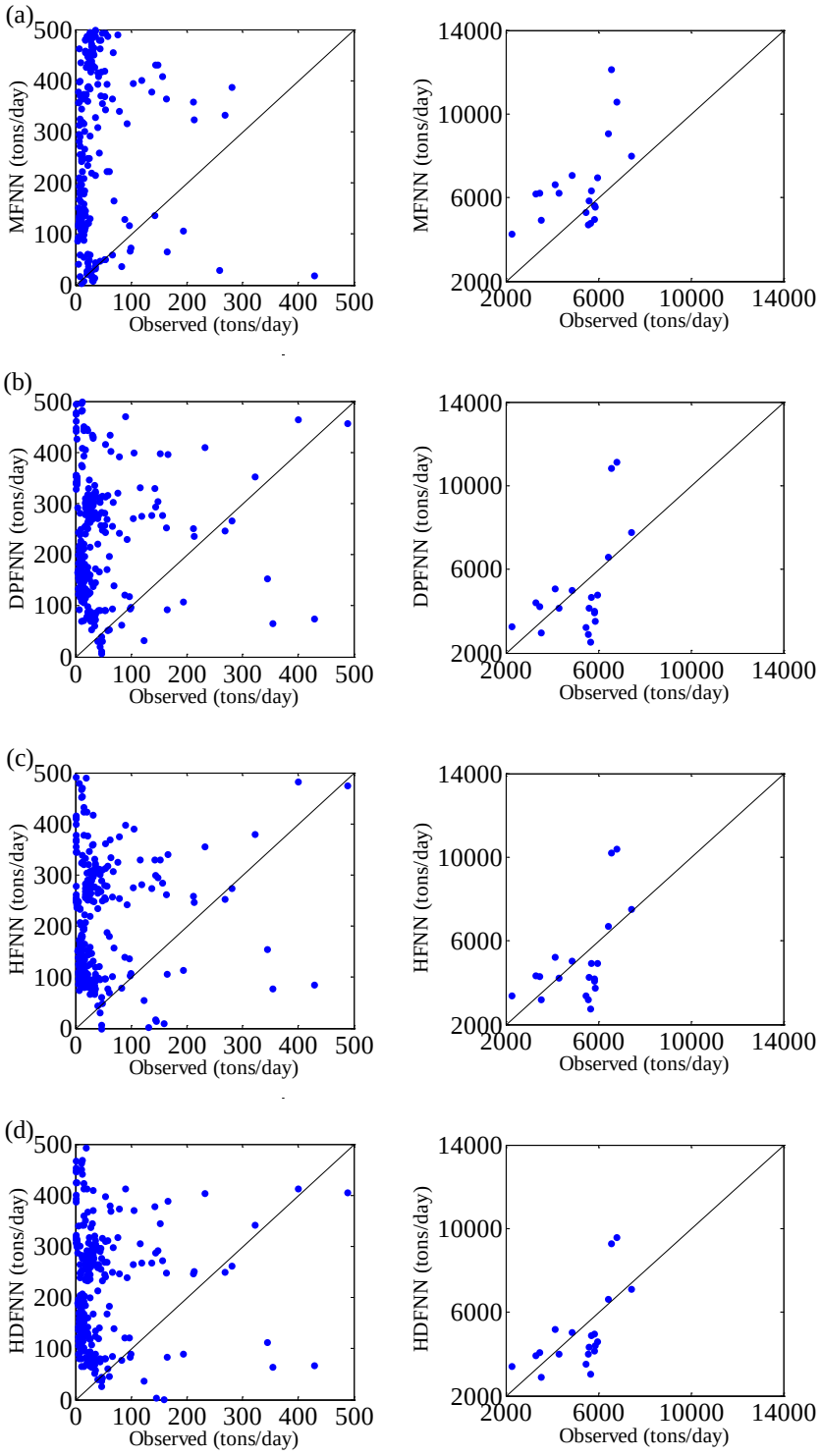

Figure 4.17 The observed and estimated suspended sediments by (a) MFNN (b) DPFNN (c) HFNN (d) HDFNN models for the study case on the Tongue River (three-day-ahead)

The values of special parameters of HFNN and HDFNN models for the present study case are presented in Table 4.11. The one-day-ahead estimation does not attain approximate values when compared with the two- and three-day-ahead cases. It is

acceptable however, since the difference is not remarkable. Besides, the a_0 and b_0 values generated by the HDFNN models are comparable with that by the HFNN models, which reveals the reliability of power relationship between discharge and sediment. Similar phenomenon could not be observed in Table 4.8 for the case on the Muddy Creek, in view of the different values of a_0 and b_0 obtained by the HFNN and HDFNN models. A possible explanation may be that in the HDFNN model the upstream sediment for the Muddy Creek case has more impacts on the final output when compared with that for the Tongue River case. Thus the effects of discharge inputs are quite different from the HFNN model with disparate parameters a_0 and b_0. The recession coefficient λ obtained in current case is generally smaller than that in the case on the Muddy Creek. The smaller recession coefficient indicates smaller storage capacity of the reservoir, and thus the output is less dependent on the upstream sediment. This exactly makes the above discussions clear. It can also explain the conflicting performances of HFNN model in Tables 4.7 and 4.10. In addition, HDFNN model with larger recession coefficient improves the dependence relation between upstream sediment and the output sediment, thus outperforms the other models more significantly. Taking the three-day-ahead case as an example, the reduction of RMSE values is respectively 44.80%, 43.28% and 42.50% when compared with that of MFNN, DPFNN and HFNN models for the case on the Muddy Creek, while the corresponding reduction is respectively 35.97%, 19.17% and 10.24% for the case on the Tongue River. In all, the recession coefficient takes effect on the storage reservoir and confirms the validation of the HDFNN model. The performances of the proposed model heavily rely on the characteristic of the studied watershed.

Table 4.11 Special parameters of HFNN and HDFNN models for the study case on the Tongue River

Input	HFNN		HDFNN		
	a_0	b_0	a_0	b_0	λ
One-day-ahead	0.3497	4.4024	0.2547	4.3120	0.5409
Two-day-ahead	0.1319	3.4968	0.2249	3.7896	0.2879
Three-day-ahead	0.1839	3.8336	0.1431	4.6848	0.3764

4.4 Summary

This chapter is concerned with the application of HDFNN model for suspended sediment load estimation. The fuzzy and time-varying characteristics of the sediment are considered in HDFNN model, while a structure of double neural networks is employed with respect to river discharge and sediment inputs. The estimates based on HDFNN models are compared with three models (namely MFNN, DPFNN and HFNN) using dataset from two study cases in Montana, USA. Main research findings are concluded as follows:

(a) The generalization and estimation abilities of HDFNN model are attested with the lowest RMSE and highest NSEC values after comparison with the other three benchmarking models. Besides, it can reproduce the medium and high loads appropriately as indicated by partitioning analysis. In view of the ability in meaningfully mapping the inputs and output, the HDFNN model could be provided as a superior alternative for *SSL* estimation.

(b) The availabilities of power relationship and storage reservoir assumption in the HFNN and HDFNN models are confirmed by parameters a_0, b_0 and recession coefficient λ, which further verifies the reliability of the proposed model.

(c) Upstream sediment is not a connatural variable with the discharge inputs, and could not be employed in the pattern classification. Double neural network is employed to

reveal different mechanisms of the inputs. The upstream sediment and discharge inputs of the HDFNN model can solely and effectively contribute the final output without contradiction.

(d) The HDFNN model presents excellent performances in multi-step-ahead estimations, particularly in high loads. The probable reason is that when the lead time is larger, the storage capacity of assumed reservoir is better displayed with larger accumulation, thus leads to better simulation in high loads.

(e) The performance of HDFNN model appears to be sensitive to studied cases. The larger recession coefficient of the case on the Muddy Creek results in better performances of HDFNN models in low loads when compared with the HFNN models. In addition, the advantage of HDFNN model over the other three models is not that distinct for the case on the Tongue River, in comparison with the Muddy Creek case. It is mainly due to smaller recession coefficient and larger drainage areas of the Tongue River case.

The present work is the first application of considering the physics embedded with the structure of double neural networks, for modeling suspended sediment series in the downstream river section. This idea for modeling *SSL* can be referred and extended for other hydrological models. Nevertheless, the reliability of the proposed HDFNN model has not been verified by uncertainty analysis, which should be performed in future work. Besides, the HDFNN model only considers the inputs of upstream discharges and sediment, and their corresponding physical mechanism. In this regard, the influences of other inputs and hydrological process involved in the sediment loads can be explored additionally to improve the model accuracy.

5 Input Determination for River Flow and Sediment Load

The previous two chapters have validated the effectiveness of two novel hybrid neural network models respectively for downstream river flow and suspended sediment load. The current chapter aims at determining the most appropriate input variables for each model, where single- and multi-step-ahead cases are taken into account concurrently. Since the possible input variables of the HNN and HDFNN models have corresponding physical meanings in relation to the river basin, the study of this chapter is an attempt to shed a light on the underlying mechanisms in the hybrid models.

5.1 Introduction

Input determination is an imperative process to identify the data-driven models. Deficient inputs without comprehensive information will bring about improper generalization. Redundant inputs that will increase the model complexity are not appreciated as well. Considering the difficulty of data measurement and scarcity of relevant data, most studies in the literature predicted river flows based on their time-series data, without the presence of spatial factors such as flows and precipitations from upstream sections (Hu et al., 2001; Wei et al., 2012; Wu et al., 2009b; Zhang, 2003). However, the discharge in downstream river is significantly affected by that in upstream rivers since practically upstream flows carry water into downstream sections in a river system. A small number of studies were carried out to predict flow at the downstream end of a river by means of the discharges or water level records of

upstream gauging sites (Chau and Cheng, 2002; Kerh and Lee, 2006; Pramanik and Panda, 2009). In particular, Wu and Chau (2006) developed a genetic algorithm-based artificial neural network model to estimate water level at the downstream station Han-Kou using input data at upstream station Luo-Shan in Yangtze River, China; Turan and Yurdusev (2009) employed feed forward back propagation neural networks, generalized regression neural network and fuzzy logic to estimate the missing values of a downstream station from those of upstream stations; rainfall–runoff modeling was undertaken by Odiyo et al. (2012) to estimate the flows that Latonyanda River contributed to Luvuvhu River downstream of Albasini Dam. These studies did not pay much attention to the influences of upstream input variables, however. In addition, mean areal precipitation over a river basin has been pronounced as a powerful influence on river flows (Bao et al., 2011; Jena et al., 2014; Johnson et al., 1999; Rezaeianzadeh et al., 2014). Given the importance to avoid flooding on the downstream side during the rainy season, it is requisite to simulate desired outflows with appropriate upstream inputs.

The input variables tend to be various in different studies for the sediment load estimation. In practice, some research employs hydro-meteorological data including rainfall as input variables (Cigizoglu and Alp, 2003; Cobaner et al., 2009) while others estimates suspended sediment with river flow and sediment data (Afan et al., 2015; Aytek and Kisi, 2008; Shamaei and Kaedi, 2016) at its own station. The upstream and downstream flow and sediment data were used as inputs by Kisi (2004b) to estimate the downstream sediment. Partal and Cigizoglu (2008) employed a combined wavelet-ANN method for downstream sediment estimation in which upstream flow data were used as possible inputs. Although the studies on the impact of upstream inputs on downstream sediment are limited, it is essential to explore the mechanism of sediment transport from upstream to downstream section.

The present chapter concentrates on the comparison of different input variables on the performances of HNN and HDFNN models. It is a supplementary study to Chapters 3 and 4, in order to determine the best inputs for the models respectively for downstream river flow and suspended sediment load. In the remaining sections of this chapter, firstly, input determination for the HNN model is performed with daily and hourly data. Then the possible input combinations for the HDFNN model are formulated with different lead times, and main findings are presented. A conclusive discussion of the input influences on the models is finally organized.

5.2 Input determination for river flow forecasting

The river at downstream section receives its flows mainly from flows and precipitations in the upstream sections. How these spatial factors influence the output flow remains an issue to explore. This section aims to figure out whether all upstream flows are essential or the nearest upstream flow is sufficient as an input variable. Additionally, the contribution of precipitations in the river basin to the modeling efficiency is examined. A comparison is made where the model with areal precipitation input is compared against that with point precipitation and without precipitation. Both daily and hourly data are applied to the forecasting model.

5.2.1 Daily river flow forecasting

5.2.1.1 Study area and data

Two study cases, namely stations along the Yellow River in the Altamaha river basin (Figure 3.3) and the North Buffalo Creek within the central Greensboro area (Figure

3.6), are employed for the daily flow forecasting. The information of stations' ID, name, location and drainage area for each case has been summarized in Tables 3.1 and Table 3.3 respectively. In order to facilitate the comparison, different variables are selected and combined as input matrixes, as demonstrated in Tables 5.1 and 5.2. For the case in the Altamaha river basin, the simplified letters *PP* and *AP* refer to point precipitation at station 02208000 and areal precipitation over the river basin, respectively; *Q*1, *Q*2 and *Q*3 are the individual flow at station 02207335, 02207220 and 02207120. As can be seen, combination 1, 2 and 3 differ from one another by the precipitation input. All three members of combination 1 do not include any precipitation input; those of combination 2 consider point precipitation as an input variable; and areal precipitation is selected in members of combination 3. In the meantime, the different members in a same group are distinguished by the different flow inputs. As an example, combination 3-1 has only one flow input *Q*1, which is from the nearest upstream section; combination 3-2 includes one more flow *Q*2; and combination 3-3 takes all the observed upstream flows into account. Thus, nine input combinations are formulated to develop the upstream-downstream flow relationship. Especially, areal precipitation over the river basin is computed by Thiessen polygon method, details of which can be found in Sen (1998). It is computed as the weighted precipitation of the four stations as described in the following equation, which can average the fluctuations recorded at each rain-gauge station.

$$AP = \sum_{i=1}^{N} w_i PP_i \tag{5.1}$$

where PP_i is the point precipitation at station i, N is the number of stations and w_i is the corresponding weight factor determined by the proportion of the basin each polygon takes up. That is to say, the weight is the relative area of each polygon, and the principle

$\sum_{i=1}^{N} w_i = 1$ is satisfied. In this study site, the value of *w* is 0.218, 0.296, 0.141 and 0.245 respectively for the four observed stations from 02207120 to 02208000. This mean areal precipitation representing effective uniform depth of rainfall over the basin is desirable owing to the uniformly distributed drainage basin. By this means, the areal precipitation is computed and plotted in Figure 5.1, together with point precipitation at station 02208000. It is notable that there are barely extreme values in the areal precipitation data, and the peak one is 0.070 m, lower than that for the point precipitation. The Thiessen polygon method produces a smooth averaging precipitation over the river basin.

For the second case in Table 5.2, the letters *PP* and *AP* refer to point precipitation at station 02095500 and areal precipitation over the river basin, respectively; *Q*1 and *Q*2 are the individual flow at station 02095271 and 02095181. The input combinations are formulated in the same manner, with different precipitation and upstream flow inputs. The areal precipitation is computed from Eq. (5.1), in which the value of *w* is 0.157, 0.233 and 0.610 respectively for the three observed stations from 02095181 to 02095500. Figure 5.2 demonstrates the areal precipitation and point precipitation at station 02095500. They are much closer to each other in comparison with the first study case, which may be due to the smaller drainage areas. In the following, the best input combination for the HNN model will be finalized based on the verification dataset optimized with DE algorithm.

Table 5.1 Input combinations with different variables in the Altamaha river basin

Combination type	Combination ID	Input variables
Without precipitation (Combination 1)	1-1	*Q*1
	1-2	*Q*1+*Q*2
	1-3	*Q*1+*Q*2+*Q*3
Point precipitation (Combination 2)	2-1	*Q*1+*PP*
	2-2	*Q*1+*Q*2+ *PP*
	2-3	*Q*1+*Q*2+*Q*3+ *PP*
Areal precipitation (Combination 3)	3-1	*Q*1+*AP*
	3-2	*Q*1+*Q*2+ *AP*
	3-3	*Q*1+*Q*2+*Q*3+ *AP*

Table 5.2 Input combinations with different variables within the central Greensboro area

Combination type	Combination ID	Input variables
Without precipitation (Combination 1)	1-1	*Q*1
	1-2	*Q*1+*Q*2
Point precipitation (Combination 2)	2-1	*Q*1+*PP*
	2-2	*Q*1+*Q*2+ *PP*
Areal precipitation (Combination 3)	3-1	*Q*1+*AP*
	3-2	*Q*1+*Q*2+ *AP*

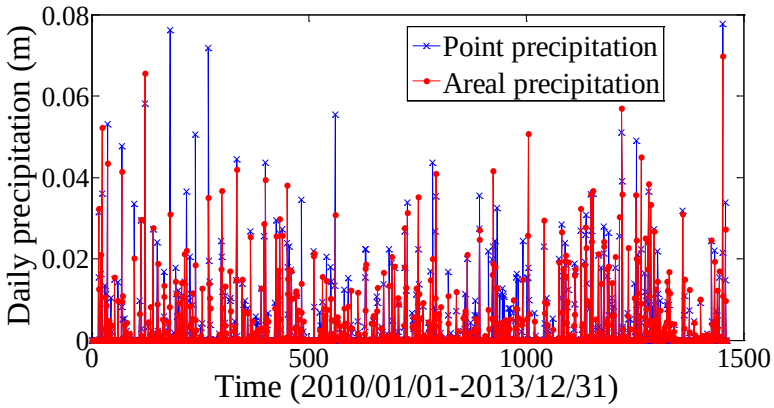

Figure 5.1 Daily point precipitation at Station 02208000 and areal precipitation over the Altamaha river basin

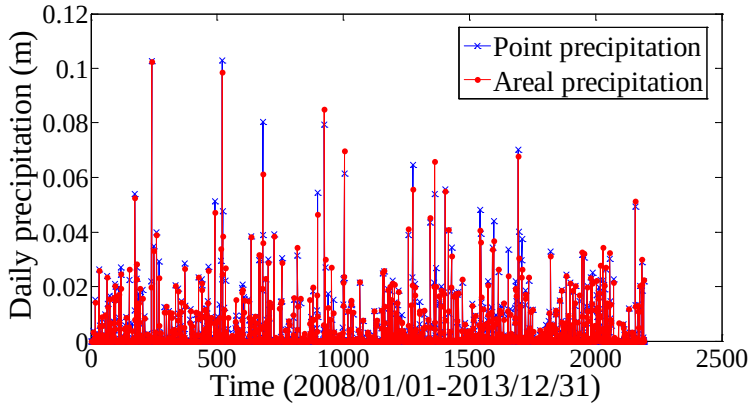

Figure 5.2 Daily point precipitation at Station 02095500 and areal precipitation within the central Greensboro area

5.2.1.2 Model performances and discussions

Three statistical measures namely RMSE, NSEC and ACC are employed to evaluate the performances of HNN model with different inputs. Both single- and multi-step-ahead forecasts are performed, and results in the testing periods are tabulated in Table 5.3. The one-day-ahead case indicates that the forecast lead time is one day, i.e., ΔT=1 day in the HNN model, and so do the two- and three-day-ahead cases. As can be seen, the results are categorized into three groups: the cases without precipitation input (combinations 1-1, 1-2 and 1-3), with point precipitation input (combinations 2-1, 2-2 and 2-3) and with areal precipitation input (combinations 3-1, 3-2 and 3-3). The values of evaluation criteria are quite different from one another in each group, indicating marked influence of upstream flows on the output. For the one-day-ahead case, combination 1-3 obtains the best RMSE and NSEC values in the first group, although failing to yield the highest ACC value. When compared with combination 1-2, there is a 12.31% reduction in RMSE value for the combination 1-3, showing its excellent performance. Thus the model fed by all three upstream flow inputs appears to be a comparatively more appreciated predictor. This conclusion is confirmed by results in the second and third groups. We can observe that combination 2-3 gives significantly superior performances in terms of the three indices among the three cases with the presence of point precipitation input. The model with input combination 3-3 achieves the lowest RMSE and highest NSEC value (i.e., 8.1500 m^3/s and 0.8294 respectively), and the ACC value is slightly smaller than that with combination 3-2. This implies that the combination 3-3 is able to reproduce favorable high values, yet the perfect fitting of overall observations is not ensured.

The observed and predicted discharges with different flow inputs are depicted in Figure 5.3. It can be perceived that most of the predicted discharges fit well with those

observations, yet the peak values are apparently over-predicted for all cases. This is acceptable for flood warning delivery, since an under-predicted forecasting would cause insufficient preparation and inaccurate precautions. Nonetheless, too much over-prediction is not appreciated as well. As indicated in Figure 5.3(a) for the inputs without precipitation, models with input combinations 1-1 and 1-2 (with cross and triangular symbols, respectively) evidently over-predict a large number of high values. Similar phenomenon can also be detected in the cases with input combinations 2-1 and 2-2 in Figure 5.3(b) as well as combinations 3-1 and 3-2 in Figure 5.3(c). A possible reason for these parsimonious predictions is that the exclusion of effective input variables $Q2$ and $Q3$ from input combination could lead to inadequate prediction. On the whole, the HNN model is sensitive to the upstream flows imposed; and generally the more information the input provides, the better the fit of the model to the observed data. It should be mentioned that it is not always proper to blindly apply all upstream flows into HNN model for other cases since the inputs from upstream sections may be redundant and unstable.

In addition to the upstream flows, the significance of the contribution of precipitation to the downstream river flow is evaluated. The scatter plot of the observed and predicted discharges in Figure 5.4 allows comparison among them. As indicated in Figure 5.4(a) with one-flow input, the one with areal precipitation does not definitely produce the best performance in comparison with the other two, since there are apparent six dots over-predicted within a range of 50-150 m^3/s. Input combination 2-1 with point precipitation outperforms others with evidence of relatively intensively distributed dots along the ideal line. This is in perfect agreement with the best RMSE and NSEC values shown in Table 5.3 when compared with combinations 1-1 and 3-1. Thus point precipitation is proven as a valuable input variable with one-flow input. In the models

with two- and three-flow inputs, it is evident from Figure 5.4 that the models without precipitation and with point precipitation perform slightly inferior to that with areal precipitation, especially for the peak and high flows. These discrepancies in the results can be attributed to the consideration of basin boundary. If the basin is limited by including only the nearest upstream section and current station, then point precipitation is adequate to yield excellent prediction. On the other hand, if all upstream flows are assigned to the HNN model, the influence of all precipitations over the river basin (represented by areal precipitation) should be contained. It is noted in Figure 5.4(b) that the input combination 3-2 gives the best performances with areal precipitation, which is the same with combination 3-3. It may be explained by that the distance between stations 02207120 and 02207220 is comparatively short, thus input combinations 3-2 and 3-3 are assumed to enjoy similar basin boundary. In short, the optimum input structure for HNN model depends on the considered basin with measured flow and precipitation data.

Table 5.3 Model performances from different input combinations in the Altamaha river basin (with daily data)

Input	one-day-ahead			two-day-ahead			three-day-ahead		
	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC
1-1	9.9994	0.7432	0.8131	10.3671	0.7246	**0.7973**	10.4562	0.7198	0.7897
1-2	11.2772	0.6733	0.7997	9.7250	0.7577	0.7839	10.3032	0.7280	0.7452
1-3	9.8891	0.7488	0.7985	10.1924	0.7338	0.7378	11.9710	0.6328	0.8004
2-1	9.0676	0.7888	0.7940	9.7049	0.7587	0.7153	11.1397	0.6820	0.7503
2-2	8.8346	0.7995	0.7989	10.6372	0.7101	0.7424	11.9307	0.6353	0.7320
2-3	8.5419	0.8126	0.8079	11.0772	0.6856	0.7674	11.5434	0.6585	0.7090
3-1	10.0161	0.7423	0.7932	10.9986	0.6900	0.6550	11.2614	0.6750	0.7134
3-2	8.5179	0.8136	**0.8142**	9.8541	0.7512	0.7051	10.7945	0.7014	0.7022
3-3	**8.1500**	**0.8294**	0.7937	**9.5636**	**0.7656**	0.6802	**10.0597**	**0.7407**	**0.8017**

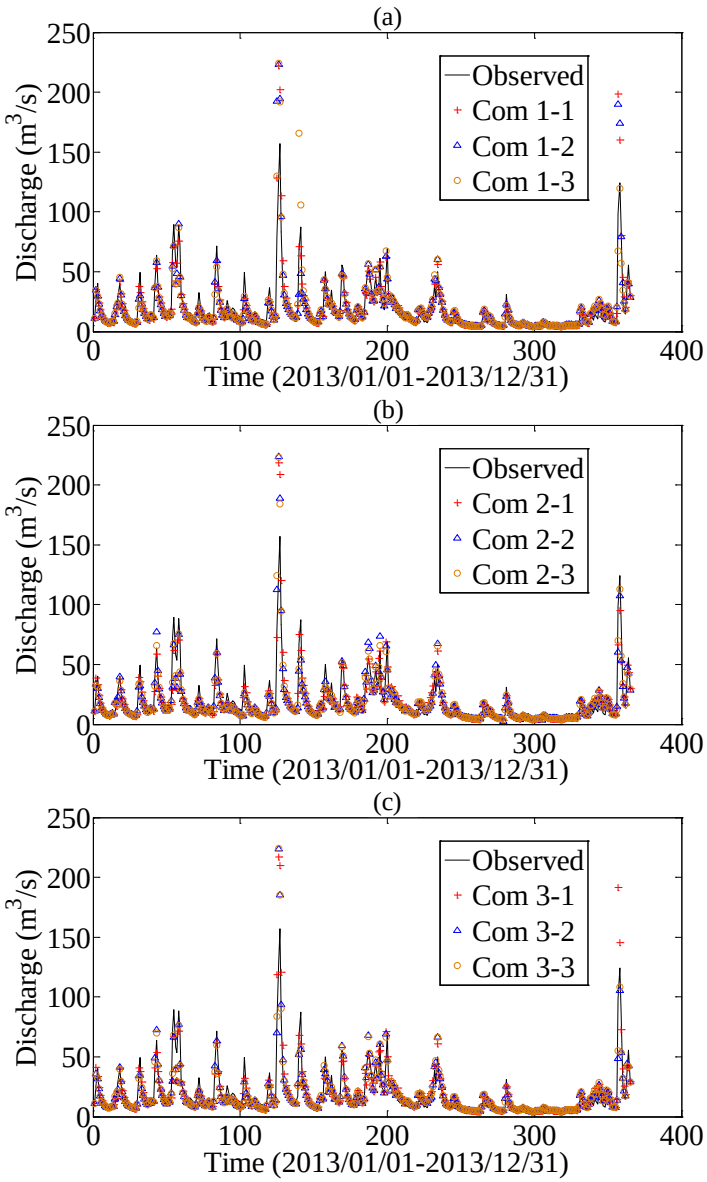

Figure 5.3 Predicted discharges (a) without precipitation (b) with point precipitation (c) with areal precipitation in the Altamaha river basin (one-day-ahead)

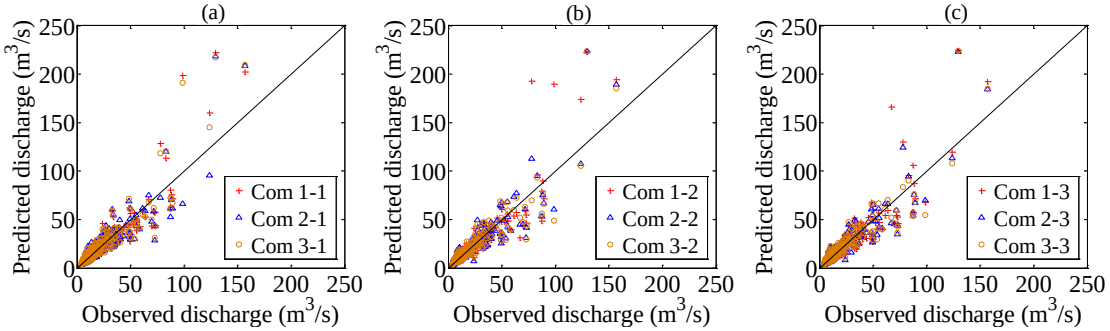

Figure 5.4 Scatter plot of observed and predicted discharges with (a) one-flow and (b) two-flow (c) three-flow input in the Altamaha river basin (one-day-ahead)

The model performances on two- and three-day-ahead forecasts from different input combinations are shown in Table 5.3. The combination 3-3 generates the lowest RMSE and highest NSEC values in both cases, although the estimations gradually deteriorate with the increase of forecast lead time. In the first input group without precipitation, the two- and three-day-ahead forecasts achieve the best results from combination 1-2 with two-flow input, which is contrary to the one-day-ahead case. Similar observations are detected in the second group with point precipitation, where the models with combination 2-1 instead of 2-3 outperform others for the two- and three-day-ahead cases. The flows at the stations 02207120 and 02207220 appear to be redundant when areal precipitation is not considered for the multi-step-ahead forecasts. This also reveals that the model performances are sensitive to the upstream flow inputs. The inclusion of all upstream flows could not definitely ensure the best forecasts, and they may increase redundancy and uncertainty with increasing lead time. Conversely, models with combination 3-3 containing all three upstream flows are appropriate since the weighted precipitation is taken as a contributing input. This areal precipitation seems to smooth the precipitations over the distributed drainage basin, and may accordingly assist to exert the influences of the corresponding upstream flows in the river basin. For the two-day-ahead case, point precipitation is a probable coupled variable with one-flow input, which accords with the one-day-ahead case. The situation is different with the three-day-ahead forecast where combination 1-1 without precipitation attains the best results in the corresponding group. A possible explanation is that when forecast lead time increases, the precipitation may not take effect timely when near upstream flows is contained as inputs. The best performances with combination 1-2 in the two-flow input group at two- and three-day-ahead forecasts further confirm the above assumption. Accordingly, the influences of input variables on model performances may be various when multi-step-ahead forecasts are performed.

Two representative parameters, namely recession coefficient and initial storage regarding the storage reservoir (representing the downstream station), are discussed finally. The values of recession coefficient computed from nine input combinations are plotted in Figure 5.5(a), and a mean value of them is given as 0.3960 for comparison. The value of recession coefficient varies within a range of 0.3609 to 0.4431, which is comparable to one another, showing the efficiency and stability of the proposed HNN model. Observation also reveals that the input combinations 1-1, 2-1 and 3-1 with only one-flow input obtain fairly high recession coefficients. This may be attributed to the fact that the storage capacity may be diminished when a large amount of flows are imported, namely, the two- and three-flow input cases. The value computed from combination 3-3 is the smallest, which also provides convincing evidence to prove the above explanation. In view of the best forecasting performances from combination 3-3, the corresponding recession coefficient of 0.3609 is regarded as an optimal one to address the storage capability of the considered reservoir.

Likewise, the values of initial storage computed from different input combinations are illustrated in Figure 5.5(b). It is defined in the same scaled interval [0, 1.0] as all the parameters. A mean value of 0.1378 is attained from the nine values ranging from 0.1298 to 0.1583. The input combination 3-3 obtains a comparatively low initial storage (i.e., 0.1327), which corresponds with the small recession coefficient**.** Accordingly, the best value of initial storage for HNN model is determined with input combination 3-3. The HNN model attains almost comparable values of initial storage with various input combinations, and the satisfactory result further confirms the validity of the HNN model with an emphasis on the time-varying storage of the river station.

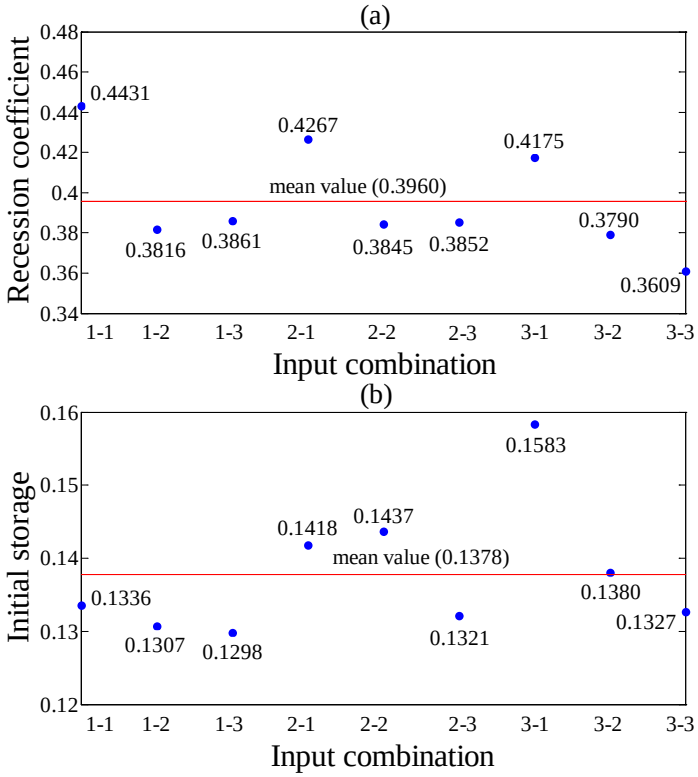

Figure 5.5 (a) Recession coefficient and (b) initial storage computed from different input combinations in the Altamaha river basin (one-day-ahead)

On account of scarcity of relevant data, two upstream flow inputs are employed in the case within the central Greensboro area. The corresponding results in terms of RMSE, NSEC and ACC are outlined in Table 5.4. As been observed in Figure 5.2, areal precipitation may be not an appropriate input variable. The time of flow travelling from one upstream section to the downstream section is relatively short, thus point precipitation may have an instant influence on the river flow. Evidence that the combination 2-2 is superior to the others for both one- and two-day-ahead forecasts convinces the above assumptions. The situation becomes reverse in the three-day-ahead case, where the areal precipitation equipping combination 3-2 is proven as an appropriate input variable. This may be attributed to the larger forecast lead time which could result in more effective collection of the precipitations over the basin.

Additionally, Figure 5.6 is demonstrated to make a comparison of different upstream

flow inputs for the one-day-ahead case. The advantages of two-flow over one-flow input are not significant in each group, and the former generally over-forecasts the high values while the latter under-forecasts them. This may be also due to the short distance between upstream and downstream sections, which leads to insensitivity of the model performances to upstream flow inputs hence. However, the two upstream flows are evaluated concurrently as effective input variables. In order to examine the effect of precipitation, scatter plots of observed and predicted discharges from different inputs are presented in Figure 5.7. Model with combination 3-1 reproduces slightly closer dots to the ideal line, particular in the low values. The observation that areal precipitation coupled with one-flow input achieves the best performances seems to be incompatible with the above results, and not agree with the conclusions from the previous study case. Plots of Figure 3.12 in Chapter 3 have implied that the daily data at Station 02095500 is relatively smooth with more small values than that at Station 02208000. This may be the reason that smoothed precipitation with one-flow can properly fit small values in the second study case. In contrast, point precipitation is a sufficiently applicable input variable together with two-flow input, as indicated in Figure 5.7(b).

Table 5.4 Model performances from different input combinations within the central Greensboro area (with daily data)

Input	one-day-ahead			two-day-ahead			three-day-ahead		
	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC
1-1	1.0858	0.8339	0.7415	1.2380	0.7844	**0.7435**	1.2429	0.7830	**0.7390**
1-2	0.9425	0.8749	**0.7833**	1.0988	0.8302	0.6961	1.1007	0.8298	0.7002
2-1	1.0846	0.8343	0.7571	1.1976	0.7982	0.6781	1.2310	0.7871	0.7308
2-2	**0.9163**	**0.8817**	0.7784	**1.0592**	**0.8422**	0.5370	1.0923	0.8324	0.6721
3-1	1.0347	0.8492	0.7717	1.1479	0.8147	0.6435	1.1683	0.8083	0.6451
3-2	0.9289	0.8784	0.7393	1.0793	0.8361	0.5585	**1.0837**	**0.8351**	0.6985

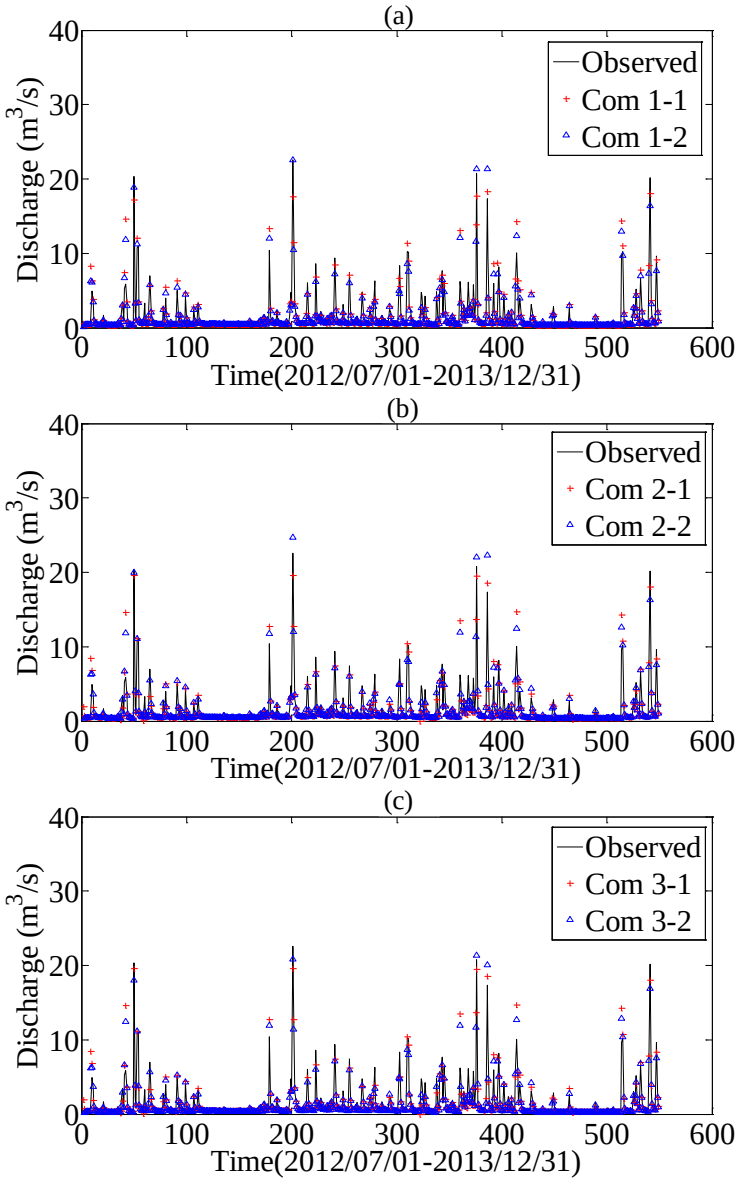

Figure 5.6 Predicted discharges (a) without precipitation (b) with point precipitation (c) with areal precipitation within the central Greensboro area (one-day-ahead)

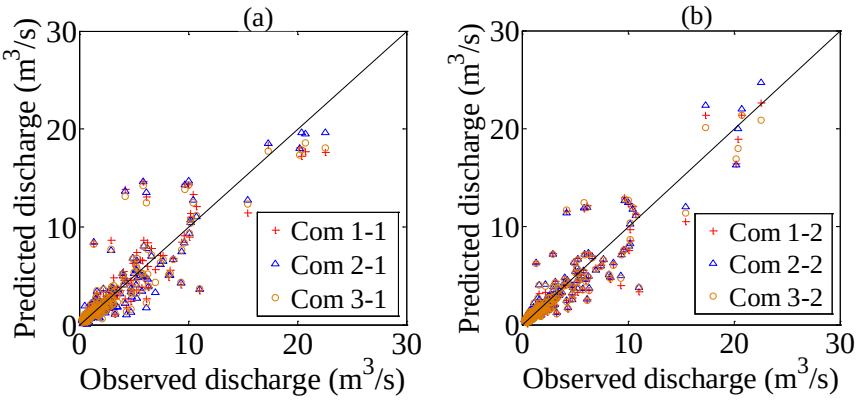

Figure 5.7 Scatter plot of observed and predicted discharges with (a) one-flow and (b) two-flow input within the central Greensboro area (one-day-ahead)

Likewise, the values of recession coefficient and initial storage generated from different inputs are outlined in Figure 5.8. They are respectively taken as 0.1951 and 0.0219 from the input combination 2-2. It seems that the models with more flow inputs are prone to generate higher values of recession coefficient, which differs from the case in the Altamaha river basin. It is reasonable that a considerably small amount of flow inputs is observed in the central Greensboro area, and they will not diminish the storage capacity as the previous study case. The values of initial storage seem to be easily susceptible to the input variables when compared with the recession coefficient, as indicated by the wide range of values varying from 0.0219 to 0.1669. It is comparable and acceptable for the HNN model however. Conclusions could be drawn from the model performances using daily data that the input variables on HNN models are highly sensitive to the area of drainage basin, features of data series and the forecast lead time.

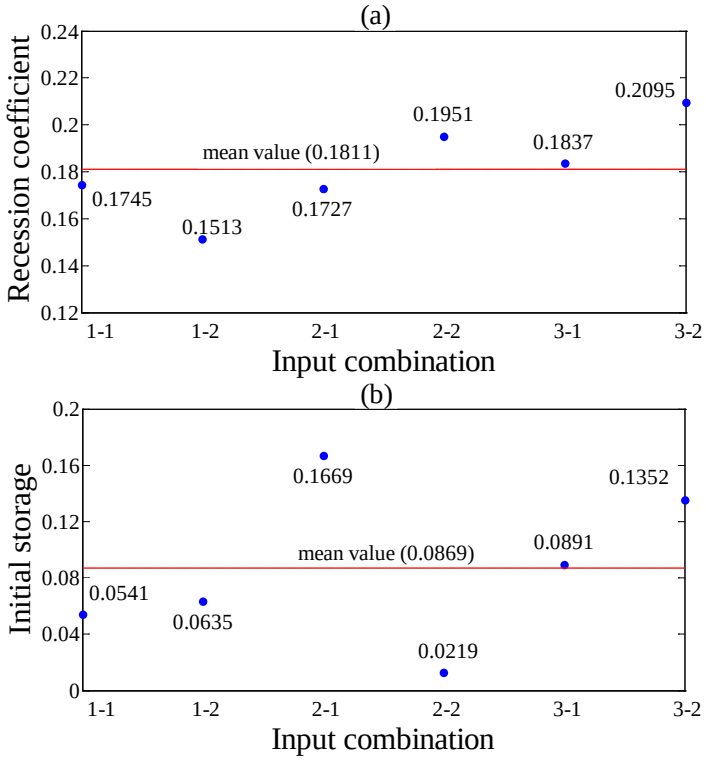

Figure 5.8 (a) Recession coefficient and (b) initial storage computed from different input combinations within the central Greensboro area (one-day-ahead)

5.2.2 Hourly river flow forecasting

5.2.2.1 Study area and data

The application of hourly data to the HNN model is employed in this section in order to determine the best input combination with different forecast lead times. The study areas in the previous section are applicable to the hourly data, which follows the procedure in Chapter 3. In this means, Tables 5.1 and 5.2 formulate practicable input combinations for the hourly river flow forecasting as well, respectively for the case in the Altamaha river basin and within the central Greensboro area. Areal precipitations for both study cases are computed in the same manner, and the comparisons with point precipitation are demonstrated in Figures 5.9 and 5.10. It is observed that the series of areal precipitation in the Altamaha river basin appears smoother than the point precipitation since it averages the fluctuations recorded. The point precipitation at station 02095500 within the central Greensboro area seems to be smaller than that at the upstream stations, thus the areal precipitation is presented to emphasize the rainfall effect over the basin. The small drainage area in the second study case may be a reason for that the upstream precipitations significantly contribute to the areal precipitation. The differences between areal and point precipitations may cause different observations of the precipitation input on the model performances, which will be discussed in details in the following.

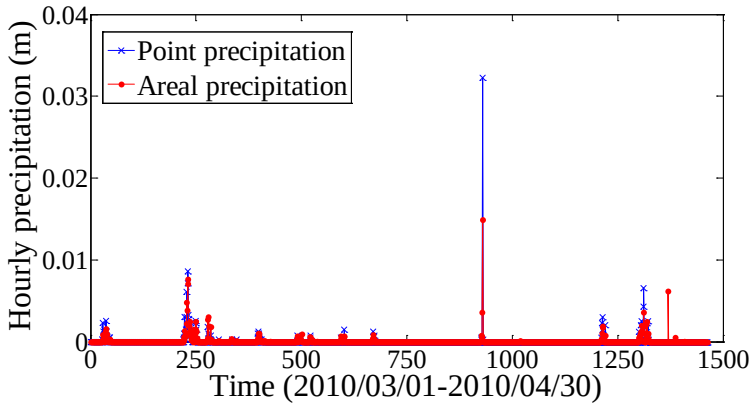

Figure 5.9 Hourly point precipitation at station 02208000 and areal precipitation over the Altamaha river basin

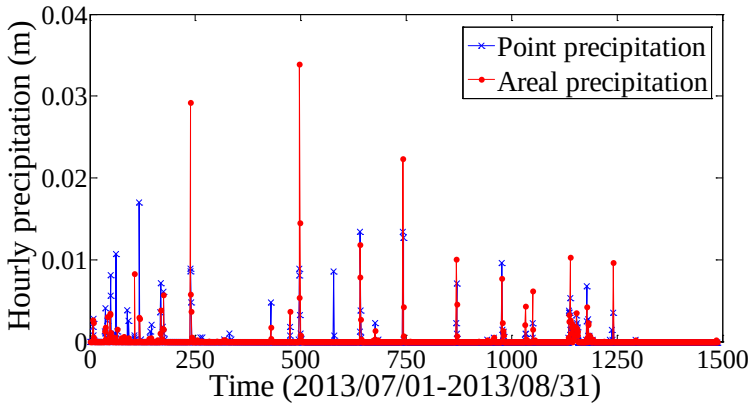

Figure 5.10 Hourly point precipitation at station 02095500 and areal precipitation within the central Greensboro area

5.2.2.2 Model performances and discussions

Regarding the case in the Altamaha river basin, the HNN model is performed with forecast lead times ΔT= 1, 4, 8 and 16 hours from different input combinations, and the results in the testing period are summarized in Tables 5.5 and 5.6. Input combination 3-3 equipped with areal precipitation and three-flow input achieves the lowest RMSE and highest NSEC and ACC values in the 1-hour-ahead case. In the first and third input groups the models with three-flow input noticeably outperform others, which reveal the efficiency of upstream flows without precipitation and with areal precipitation. The one-flow input is appropriately coupled with point precipitation, as indicated by the best performances from combination 2-1.

Figure 5.11 is plotted to provide visual inspection of the influences of upstream flows on the model performances. As can be seen, the models without precipitation could not basically fit the low values, and the combination 1-3 tends to better approximate the high values than the other two. As revealed in Figure 5.11(b), both combination 2-1 and 2-3 generate satisfactory values of the valley, yet, the former is superior to the latter in the observations below 20 m^3/s. Model fed by combination 3-3 matches the observations best, although under-forecasts the peak value. Besides, the model

performances from different precipitation input are demonstrated in Figure 5.12. The values attained by combination 2-1 distribute much closer than the other two, implying that point precipitation has a fair cooperation with the one-flow input. On the contrary, the areal precipitation shows its superiority with the two- and three-flow inputs. These results are reasonable in accordance with those from the daily forecasting, that the impacts of inputs strongly rely on the consideration of basin boundary.

Table 5.5 Model performances from different input combinations in the Altamaha river basin (with 1- and 4-hour-ahead forecasting)

Input combination	1-hour-ahead			4-hour-ahead		
	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC
1-1	4.2186	0.7537	0.7367	4.2760	0.7489	0.6600
1-2	3.2826	0.8509	0.7436	4.0415	0.7757	0.6972
1-3	3.0201	0.8738	0.7636	4.0462	0.7752	0.7329
2-1	1.9096	0.9495	0.8585	2.8660	0.8872	0.8051
2-2	2.5392	0.9108	0.8829	3.1187	0.8664	**0.8989**
2-3	2.0142	0.9439	0.9168	3.1441	0.8643	0.8332
3-1	2.8809	0.8851	0.7989	**2.3604**	**0.9235**	0.8842
3-2	1.8395	0.9532	0.9138	2.7485	0.8963	0.8890
3-3	**1.7544**	**0.9574**	**0.9200**	2.8259	0.8903	0.8836

Table 5.6 Model performances from different input combinations in the Altamaha river basin (with 8- and 16-hour-ahead forecasting)

Input combination	8-hour-ahead			16-hour-ahead		
	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC
1-1	4.3063	0.7479	0.6650	4.6453	0.7125	0.6727
1-2	4.7812	0.6893	0..6679	5.1578	0.6455	0.6691
1-3	3.8083	0.8029	0.7205	5.3633	0.6167	0.6778
2-1	**2.4746**	**0.9168**	0.8526	**2.8348**	**0.8929**	0.8643
2-2	3.2063	0.8603	**0.8780**	3.4783	0.8388	**0.8881**
2-3	3.1289	0.8669	0.8478	3.7167	0.8159	0.8570
3-1	2.5643	0.9106	0.8642	3.0348	0.8773	0.8622
3-2	3.0620	0.8726	0.8385	3.8681	0.8006	0.8311
3-3	2.9499	0.8817	0.8651	3.8208	0.8055	0.8062

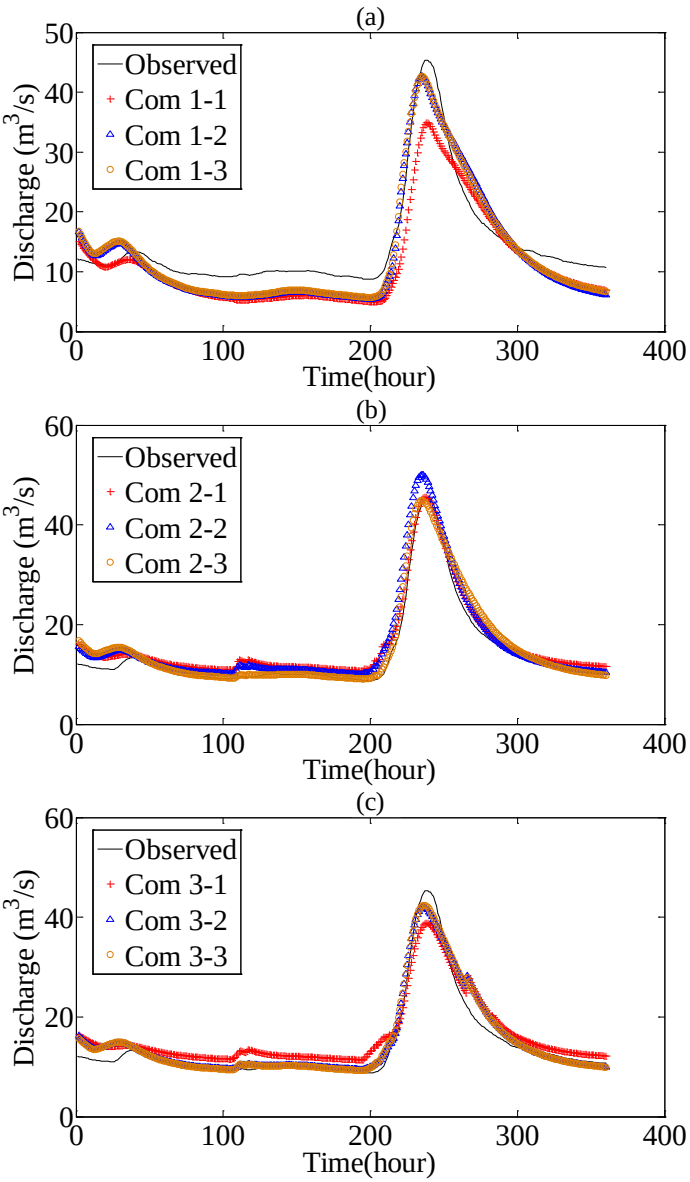

Figure 5.11 Predicted discharges (a) without precipitation (b) with point precipitation (c) with areal precipitation in the Altamaha river basin (one-hour-ahead)

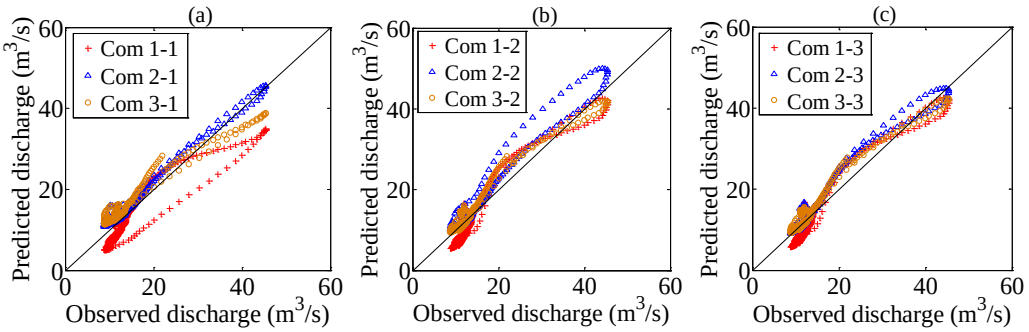

Figure 5.12 Scatter plot of observed and predicted discharges with (a) one-flow and (b) two-flow (c) three-flow input in the Altamaha river basin (one-hour-ahead)

The contributions of input variables to the downstream river flow may perform differently with respect to multi-step-ahead forecasts. As seen from Tables 5.5 for the 4-hour-ahead case, input combination 3-1 yields the best RMSE and NSEC values. The flows at station 02207100 and 02207200 seem to be superfluous in comparison with the single-step-ahead case. Furthermore, the 8- and 16-hour-ahead forecasts tend to favor the point precipitation and one-flow input, as indicated by the best performances with combination 2-1 in Tables 5.6. The precipitations and flows at station 02207100 and 02207200 may trigger a redundant and uncertain generalization with increasing lead time. It is noted that the daily and hourly cases achieve the best performances from different input combinations in the multi-step-ahead cases. The inconsistent performances lie behind the differences between flow travel time and forecast lead time. The estimated time of flow travelling from station 02207335 and 02208000 is about 16 hours. As for the daily forecasts, the lead time (i.e., 1, 2 and 3 day) is larger than travel time, thus the flows and precipitations in the upstream stations are fully collected and serve as effective inputs for the downstream flow. On the other hand, the lead times chosen in the hourly forecast are within the travel time, where the stability of the mechanism of mapping the inputs and output could not be ensured. As a result, when the forecast lead time increases the models with combination 3-3 consistently achieves best daily forecasts, whereas the hourly upstream flows and areal precipitation are undesirable inputs for the downstream flow.

The values of recession coefficient and initial storage concerning the one-hour-ahead forecast in the Altamaha river basin are illustrated in Figure 5.13. The optimal ones are determined from input combination 3-3, with value of 0.9439 and 0.2980 respectively. They are nearly approximate to each mean value computed from nine input combinations, which would be a concrete evidence of the stability of HNN model for

the hourly forecasting in this study area. Another interesting phenomenon can be observed is that the recession coefficient generated by hourly forecast is generally larger than that by the daily forecast. The assumed reservoir seems to play larger storage capacity from the time dimension of hour.

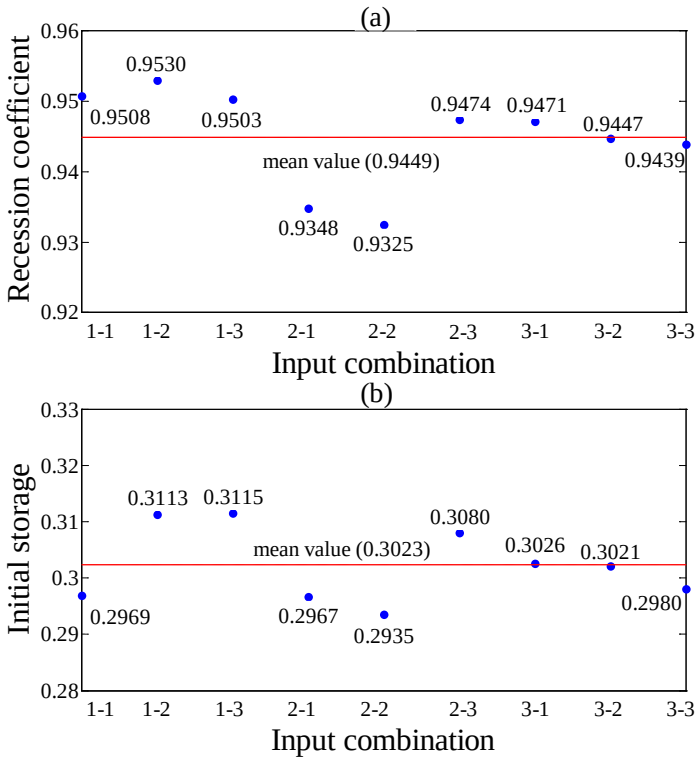

Figure 5.13 (a) Recession coefficient and (b) initial storage computed from different input combinations in the Altamaha river basin (one-hour-ahead)

As for the case within the central Greensboro area, hourly forecasts are performed with the same forecast lead times, results of which in terms of RMSE, NSEC and ACC are presented in Tables 5.7 and 5.8. As can be observed, precipitation input could not show its efficiency on model output at 1-hour-ahead case. The improvements of combination 1-2 over 2-2 and 3-2 are respectively 1.65% and 0.46% in terms of NSEC value. Besides, models with two-flow input are superior to that with one-flow, and the reductions in RMSE value are 6.43%, 17.43% and 6.76% respectively for the cases without precipitation, with point precipitation and with areal precipitation. The comparisons are visually presented in Figure 5.14, in order to shown the influences of

upstream flows. Both combinations 1-1 and 1-2 under-forecast the peak values of the first and forth valley in the time series, and they perfectly fit the observations of the third valley. The former attains slightly larger peak value and some high values than the latter, which finally results in its worse performances in terms of RMSE and NSEC. The model with combination 2-1 reproduces improper high values when compared with its counterpart, as indicated in Figure 5.14(b). The two-flow input with areal precipitation does not show great superiority over the one-flow input, revealing the weakness of flow input effect on the output. In addition, the influences of precipitation input are presented in Figure 5.15. In the group of combinations equipped with one-flow input, the performances attained from different precipitation input show considerable differences from one another. The combination 2-1 over-forecasts most of the observations, indicating the unpromising effect of point precipitation. The distributions of dots reproduced from inputs without precipitation are slightly closer to the ideal line than the other two in Figure 5.15(b), which confirms the unavailability of precipitation.

The above conclusions can also be drawn from the results with 4-hour-ahead forecast, where combination 1-2 achieves the best results in view of the lowest RMSE and highest NSEC values. The best ACC value is attained from combination 3-2, although having inferior performances on reproducing high values. This also reveals that the two-flow input plays steadily effective role on the HNN model. The 4-hour-ahead forecast yields acceptable results from the perspective of practical application, since the NSEC and ACC values obtained by the best input are respectively as high as 0.8727 and 0.7749. Nevertheless, the forecasts with 8- and 16-hour-ahead seem to be inapplicable as indicated in Table 5.8. These results could tell certain facts however, that the precipitation coupled with two-flow input begins to function with increasing lead time. The possible reasons have been stated in the previous study case that when forecast lead

times chosen are within the travel time (estimated by 8 hours in current case), the mapping of the inputs and output may be unstable and not every input variable at the upstream station could effectively work on the downstream flow.

Table 5.7 Model performances from different input combinations within the central Greensboro area (with 1- and 4-hour-ahead forecasting)

Input combination	1-hour-ahead			4-hour-ahead		
	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC
1-1	1.6597	0.8558	**0.8038**	1.8476	0.8213	0.4870
1-2	**1.5530**	**0.8737**	0.7944	**1.5596**	**0.8727**	0.7749
2-1	1.9845	0.7938	0.5241	1.9945	0.7918	0.3941
2-2	1.6385	0.8595	0.7542	1.6486	0.8577	0.7100
3-1	1.6922	0.8501	0.7650	1.7073	0.8474	0.6934
3-2	1.5778	0.8697	0.7824	1.6180	0.8629	**0.7804**

Table 5.8 Model performances from different input combinations within the central Greensboro area (with 8- and 16-hour-ahead forecasting)

Input combination	8-hour-ahead			16-hour-ahead		
	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC
1-1	2.5863	0.6523	0.5134	3.6458	0.3155	0.2678
1-2	2.6015	0.6482	0.4580	2.6021	0.6480	**0.5194**
2-1	2.4938	0.6767	0.4321	2.5218	0.6694	0.4427
2-2	**2.4061**	**0.6990**	0.4344	**2.4360**	**0.6915**	0.3435
3-1	2.6091	0.6461	0.3716	2.6100	0.6459	0.3969
3-2	2.5468	0.6628	**0.5050**	2.5593	0.6595	0.3014

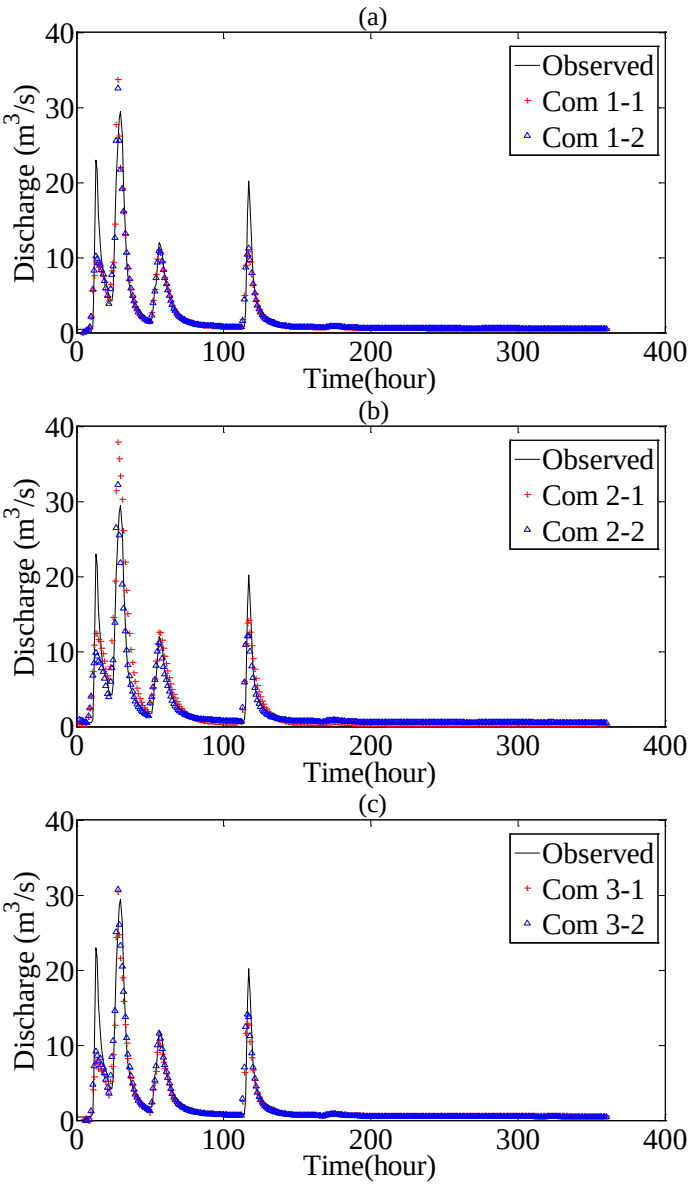

Figure 5.14 Predicted discharges (a) without precipitation (b) with point precipitation (c) with areal precipitation within the central Greensboro area (one-hour-ahead)

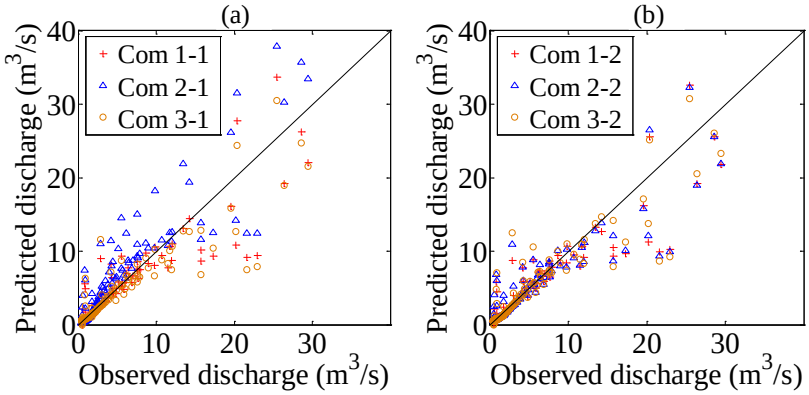

Figure 5.15 Scatter plot of observed and predicted discharges with (a) one-flow and (b) two-flow input within the central Greensboro area (one-hour-ahead)

Similarly, the values of recession coefficient and initial storage computed from different input combinations are plotted in Figure 5.16. The recession coefficient for the hourly forecast (i.e., 0.6754) is larger than the corresponding daily one, in agreement with the case in the Altamaha river basin. Mean initial storage is computed as 0.1524 from the values within the range between 0.1207 and 0.1956. The initial storage tends to be more sensitive to the input variables in comparison with the recession coefficient, for both daily and hourly forecasts in the two study cases. It is rational since the initial storage discussed herein is scaled within a particular range, but in fact, it has practical value (with unit m^3) corresponding to the inputs with different values of data.

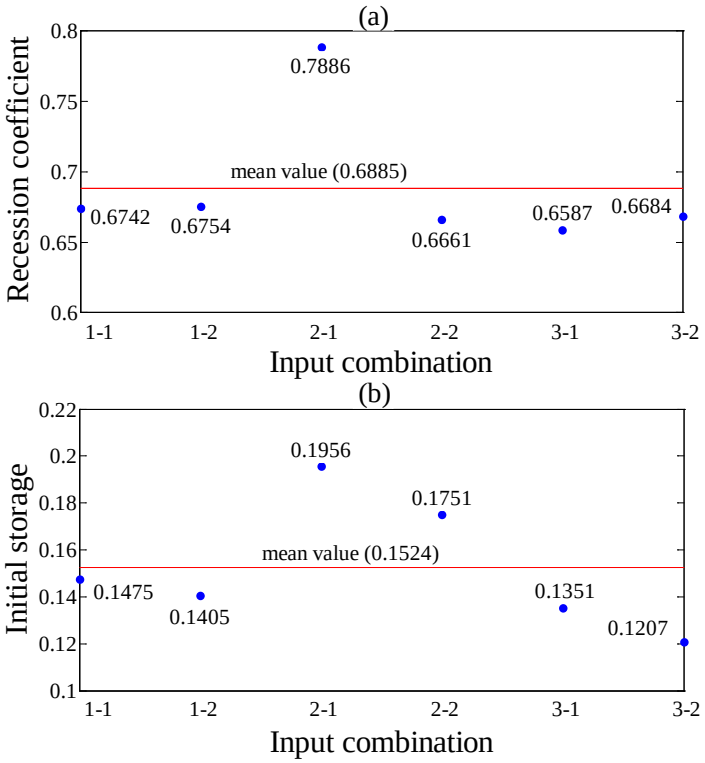

Figure 5.16 (a) Recession coefficient and (b) initial storage computed from different input combinations within the central Greensboro area (one-hour-ahead)

As a whole, the functions of upstream flow and precipitation as input variables strongly depend on the forecast lead time for the hourly cases. When forecast lead times are shorter than the travel time, certain input variables including the precipitation may be

uncertain and redundant for the HNN model. Multi-step-ahead forecasting is prone to be better performed in the Altamaha river basin when compared with that within the Greensboro area, with the evidence of the high ACC value (i.e., 0.8643) even when the lead time is 16 hour. This may be owing to the less fluctuant data series and larger recession coefficient in the former study case.

5.3 Input determination for sediment load estimation

This section attempts to determine effective input variables for the HDFNN model proposed in Chapter 4. The same study areas are employed with daily data during the same time period. Both single- and multi-step-ahead estimations are performed with various input combinations, and the results are discussed in the following.

5.3.1 Study area and data

Stations 06088300 and 06088500 on the Muddy Creek in Montana USA as well as stations 06307830 and 06308500 on the Tongue River are chosen as study cases as before, and their locations on the map are observed in Figures 4.2 and 4.4. The relations of sediment and flow data at both upstream and downstream stations have been plotted in Figures 4.3 and 4.5 respectively for the two cases, with data at the same time step. Since input variables at different time may have different influences on the output, the cross-correlations between the data are addressed before the estimation. Table 5.9 summaries the cross-correlations between the SSL_u, Q_u, Q_d data at different time steps and SSL_d data at time t for the first study case. As expected, the highest correlations are found between each input variable and the output at the same time. For instance, the cross-correlation between $SSL_{u(t)}$ and $SSL_{d(t)}$ reaches as high as 0.9064. However, the

degrees of relation gradually decrease with inputs more ahead of time. When the lead time is given as 4 day, the cross-correlations are reduced to 0.1552, 0.2361 and 0.2443 respectively for the SSL_u, Q_u, Q_d inputs. Similarly, the cross-correlations between the input variables and downstream sediment load for the case on the Tongue River are presented in Table 5.10. The reductions of cross-correlation with lead time seem to be mitigated when compared with the values in Table 5.9, which may finally result in different influences of input variables on the output in each study case specifically for the multi-step-ahead estimations. In contrast to the previous study case, cross-correlations between Q_d and SSL_d are consistently larger than that between the Q_u and SSL_d particularly. It is reasoned by the larger drainage areas in the Tongue River case that the upstream flows tend to have fewer influences on the downstream sediment.

Table 5.9 The cross-correlation of the data of the stations on the Muddy Creek

	$SSL_{d(t)}$		$SSL_{d(t)}$		$SSL_{d(t)}$
$SSL_{u(t)}$	0.9064	$Q_{u(t)}$	0.7287	$Q_{d(t)}$	0.6878
$SSL_{u(t-1)}$	0.6911	$Q_{u(t-1)}$	0.6047	$Q_{d(t-1)}$	0.5590
$SSL_{u(t-2)}$	0.6033	$Q_{u(t-2)}$	0.4876	$Q_{d(t-2)}$	0.4711
$SSL_{u(t-3)}$	0.3993	$Q_{u(t-3)}$	0.3741	$Q_{d(t-3)}$	0.3692
$SSL_{u(t-4)}$	0.1552	$Q_{u(t-4)}$	0.2361	$Q_{d(t-4)}$	0.2443

Table 5.10 The cross-correlation of the data of the stations on the Tongue River

	$SSL_{d(t)}$		$SSL_{d(t)}$		$SSL_{d(t)}$
$SSL_{u(t)}$	0.7749	$Q_{u(t)}$	0.5967	$Q_{d(t)}$	0.7424
$SSL_{u(t-1)}$	0.6444	$Q_{u(t-1)}$	0.5621	$Q_{d(t-1)}$	0.6329
$SSL_{u(t-2)}$	0.5433	$Q_{u(t-2)}$	0.5040	$Q_{d(t-2)}$	0.5289
$SSL_{u(t-3)}$	0.5077	$Q_{u(t-3)}$	0.4542	$Q_{d(t-3)}$	0.4884
$SSL_{u(t-4)}$	0.4880	$Q_{u(t-4)}$	0.4214	$Q_{d(t-4)}$	0.4744

In this section, estimations with one-, two- and three-day-ahead upstream sediment input are undertaken for each study case, wherein the inputs Q_u, Q_d could be chosen with different ahead of times. The abnormity of input variable SSL_u is attributed to the mechanism of the HDFNN model in which the output $SSL^{(2)}$ at time $T+\Delta T$ is computed from the upstream sediment at time T as indicated by Eqs. (4.10) and (4.11). As a result,

the upstream sediment applied to all possible input combinations is set to $SSL_{u(t-1)}$ for the one-day-ahead estimation. Those for the two- and three-day-ahead cases are respectively $SSL_{u(t-2)}$ and $SSL_{u(t-3)}$ in the same manner. It should be noted that the influence of upstream sediment is as important as the upstream and downstream flows. However, it fails to be implemented in this chapter owing to the unavailability of data collection at other upstream stations within the studied basins. Similar endeavor has been undertaken in the HNN model that the impacts of upstream flows on the downstream flow are examined. Representative input combinations with different lead times are formulated in Table 5.11. The main objective of this section is to compare the influences of Q_u, Q_d with different ahead of times and determine appropriate input variables for the HDFNN model.

Table 5.11 The input combinations with different variables for the sediment estimations

	Input ID	Input variables
One-day-ahead (Combination 1)	1-1	$[Q_{u(t-1)}, Q_{d(t)}, SSL_{u(t-1)}]$
	1-2	$[Q_{u(t)}, Q_{d(t)}, SSL_{u(t-1)}]$
	1-3	$[Q_{u(t-1)}, Q_{u(t)}, Q_{d(t)}, SSL_{u(t-1)}]$
	1-4	$[Q_{u(t-1)}, Q_{d(t)}, Q_{d(t-1)}, SSL_{u(t-1)}]$
	1-5	$[Q_{u(t)}, Q_{d(t)}, Q_{d(t-1)}, SSL_{u(t-1)}]$
	1-6	$[Q_{u(t-1)}, Q_{u(t)}, Q_{d(t)}, Q_{d(t-1)}, SSL_{u(t-1)}]$
Two-day-ahead (Combination 2)	2-1	$[Q_{u(t-1)}, Q_{u(t)}, Q_{d(t)}, SSL_{u(t-2)}]$
	2-2	$[Q_{u(t-1)}, Q_{d(t)}, Q_{d(t-1)}, SSL_{u(t-2)}]$
	2-3	$[Q_{u(t-1),} Q_{u(t)}, Q_{d(t)}, Q_{d(t-1)}, SSL_{u(t-2)}]$
	2-4	$[Q_{u(t)}, Q_{d(t)}, Q_{d(t-1)}, Q_{d(t-2),} SSL_{u(t-2)}]$
	2-5	$[Q_{u(t-1),} Q_{u(t)}, Q_{d(t)}, Q_{d(t-1)}, Q_{d(t-2),} SSL_{u(t-2)}]$
	2-6	$[Q_{u(t-2),} Q_{u(t-1),} Q_{u(t)}, Q_{d(t)}, Q_{d(t-1)}, Q_{d(t-2),} SSL_{u(t-2)}]$
Three-day-ahead (Combination 3)	3-1	$[Q_{u(t-1),} Q_{u(t)}, Q_{d(t)}, Q_{d(t-1)}, SSL_{u(t-3)}]$
	3-2	$[Q_{u(t-1),} Q_{u(t)}, Q_{d(t)}, Q_{d(t-1)}, Q_{d(t-2),} SSL_{u(t-3)}]$
	3-3	$[Q_{u(t-2),} Q_{u(t-1),} Q_{u(t)}, Q_{d(t)}, Q_{d(t-1)}, Q_{d(t-2),} SSL_{u(t-3)}]$
	3-4	$[Q_{u(t-1),} Q_{u(t)}, Q_{d(t)}, Q_{d(t-1)}, Q_{d(t-2),} Q_{d(t-3),} SSL_{u(t-3)}]$
	3-5	$[Q_{u(t-2),} Q_{u(t-1),} Q_{u(t)}, Q_{d(t)}, Q_{d(t-1)}, Q_{d(t-2),} Q_{d(t-3),} SSL_{u(t-3)}]$
	3-6	$[Q_{u(t-3),} Q_{u(t-2),} Q_{u(t-1),} Q_{u(t)}, Q_{d(t)}, Q_{d(t-1)}, Q_{d(t-2),} Q_{d(t-3),} SSL_{u(t-3)}]$

5.3.2 Model performances and discussions

As far as the case on the Muddy Creek is concerned, the model performances in terms of RMSE and NSEC with one-day-ahead estimation are primarily demonstrated in Table 5.12. Input combination consisting of $Q_{u(t\text{-}1)}$, $Q_{u(t)}$, $Q_{d(t)}$, $Q_{d(t\text{-}1)}$ and $SSL_{u(t\text{-}1)}$ achieves the best results, where the NSEC value reaches as high as 0.9762. The reductions in RMSE value with combination 1-3 are respectively 31.70% and 20.07% when compared with the 1-1 and 1-2. These three combinations enjoy a same Q_d input, thus great influences of upstream flow are accordingly revealed. Both $Q_{u(t\text{-}1)}$ and $Q_{u(t)}$ variables seem to be indispensable for the one-day-ahead estimation. The partitioning analysis is employed as well to examine the model performances in different loads, as shown in Table 5.12. The combination 1-6 better approximate the medium and high loads, which still implies its efficiency in the estimation. It is observed in Table 5.13 that the two-day-ahead case attains the lowest RMSE and highest NSEC values from combination 2-3, whose flow inputs are identical to the 1-6. The inclusion of $Q_{d(t\text{-}2)}$ in combinations 2-4, 2-5 and 2-6 appears to be redundant. In addition, combination 2-3 is better than 2-1 merely for the estimation of high values whereas its RMSE values in low and medium loads are relatively large. It means that the $Q_{d(t\text{-}1)}$ input variable functions on increasing the accuracy of fitting high values.

When examining the three-day-ahead estimation in Table 5.14, input combination 3-1 equipped with the same flow inputs obtains the highest NSEC value, suggesting that the flow variables two- and three-day-ahead of time either on the upstream or downstream stations could not contribute to the downstream sediment. It is reasonable since the cross-correlations between flow input and output reduce to low values when the lead times are more than one day. Besides, the inclusion of input variables $Q_{u(t\text{-}2)}$ and $Q_{d(t\text{-}2)}$ in combination 3-3 strengthens the model ability in reproducing low loads, while the

combination 3-4 shows advantage in matching medium loads. The inconformity of performances of input combinations on different loads may not be generalized. However, it reveals that the influences of input variables in terms of different ahead of times could be more complicated and uncertain with increasing lead time.

Table 5.12 Model performances from different input combinations for the study case on the Muddy Creek (one-day-ahead)

Input combination	Total RMSE (tons/day)	NSEC	Low RMSE (tons/day)	Medium RMSE (tons/day)	High RMSE (tons/day)
1-1	564.0116	0.9425	**70.3698**	455.9110	1139.708
1-2	481.9468	0.9580	95.7952	176.7717	1057.852
1-3	385.2324	0.9732	100.1694	171.8181	830.3655
1-4	485.8397	0.9573	83.1704	211.7877	1059.875
1-5	394.6522	0.9718	99.6747	170.3804	853.3354
1-6	**362.8466**	**0.9762**	112.2258	**161.2308**	**776.1632**

Table 5.13 Model performances from different input combinations for the study case on the Muddy Creek (two-day-ahead)

Input combination	Total RMSE (tons/day)	NSEC	Low RMSE (tons/day)	Medium RMSE (tons/day)	High RMSE (tons/day)
2-1	407.1043	0.9703	**87.1773**	**146.3615**	891.5690
2-2	417.2932	0.9686	96.1004	156.5874	910.4079
2-3	**384.6389**	**0.9733**	113.8330	172.2728	**823.0381**
2-4	417.6001	0.9685	100.8666	156.5884	909.7979
2-5	488.6829	0.9569	107.2553	206.3164	1061.114
2-6	901.6060	0.8534	100.7966	167.3921	2021.598

Table 5.14 Model performances from different input combinations for the study case on the Muddy Creek (three-day-ahead)

Input combination	Total RMSE (tons/day)	NSEC	Low RMSE (tons/day)	Medium RMSE (tons/day)	High RMSE (tons/day)
3-1	**402.3409**	**0.9709**	101.8828	169.0317	**868.8377**
3-2	466.0711	0.9609	109.0735	172.5343	1015.545
3-3	762.7784	0.8953	**70.0881**	246.4649	1691.351
3-4	462.0712	0.9612	113.4957	**135.2544**	1013.654
3-5	652.0077	0.9235	96.6598	171.2376	1448.626
3-6	730.3532	0.9040	87.9750	217.7262	1620.819

More insights on the models performances from different input combinations can be obtained through a visual inspection of the scatter plots of data below 2000 tons/day. As in Figure 5.17 where the downstream flow is fixed as [$Q_{d(t)}$, $Q_{d(t\text{-}1)}$], the effects of upstream flows could be observed by performances from combinations 1-4, 1-5 and 1-6. The model with $Q_{u(t\text{-}1)}$ over-estimates many observations while that with $Q_{u(t)}$ tends to be an under-estimator. The superiority of combination 1-6 over the 1-5 is not obvious than the 1-4 from the figures, which may lead to a conclusion that the $Q_{u(t)}$ plays a more important role than the $Q_{u(t\text{-}1)}$. However, combination including both of them assures best estimation for the output. Figure 5.18 is plotted to show the influences of downstream flows with fixed upstream flows [$Q_{u(t)}$, $Q_{u(t\text{-}1)}$] for the two-day-ahead estimation. It seems that the distributions of computed dots from the three combinations do not markedly differ from each other, thus the downstream flow inputs may not have crucial influences on the loads lower than 2000 tons/day. Similar findings are observed in Figure 5.19 with same upstream flow inputs for the three-day-ahead estimation. This also implies that the impacts of downstream flow inputs are mainly on the high loads, which generate quite different RMSE values in high loads as indicated in Tables 5.13 and 5.14. The model performances with various upstream flow inputs for the three-day-ahead estimation are demonstrated in Figures 5.20. The additions of $Q_{u(t\text{-}2)}$ and $Q_{u(t\text{-}3)}$ in the combinations 3-5 and 3-6 render wider spread dots visibly, indicating the unpromising functions of upstream flows ahead of two and three days. When comparing the model performances between Figures 5.19 and 5.20, the upstream flows deliver much more differences than the downstream flows on the output, which might be reasoned by the higher cross-correlation between the Q_u and SSL_d.

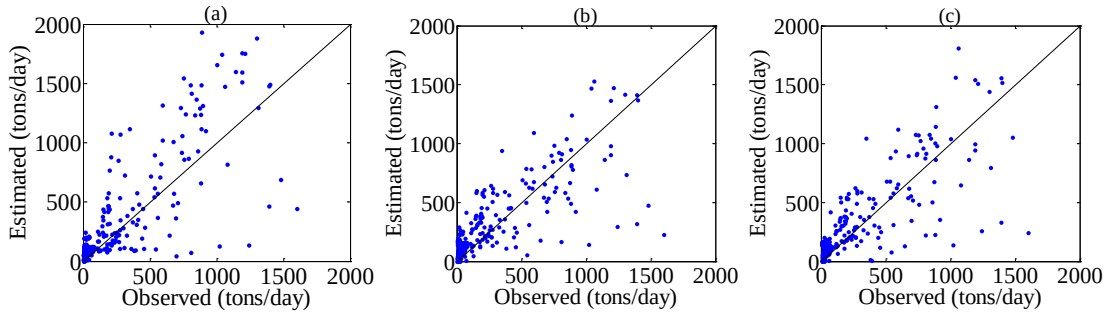

Figure 5.17 Scatter plot of observed and estimated sediment from input combinations (a) 1-4, (b) 1-5 and (c) 1-6 on the Muddy Creek

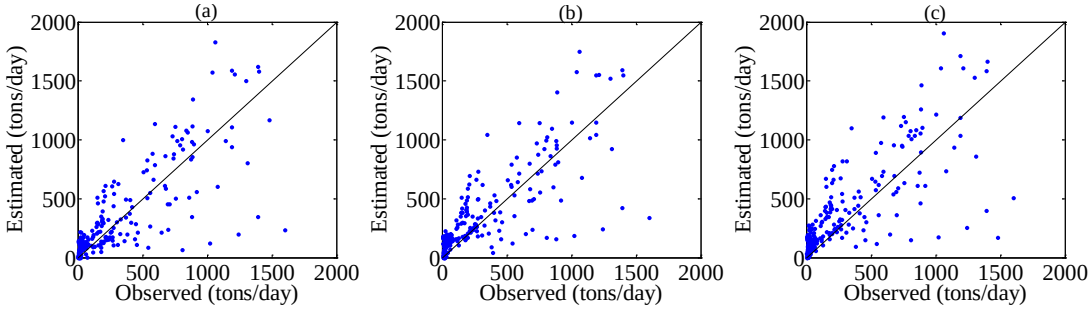

Figure 5.18 Scatter plot of observed and estimated sediment from input combinations (a) 2-1, (b) 2-3 and (c) 2-5 on the Muddy Creek

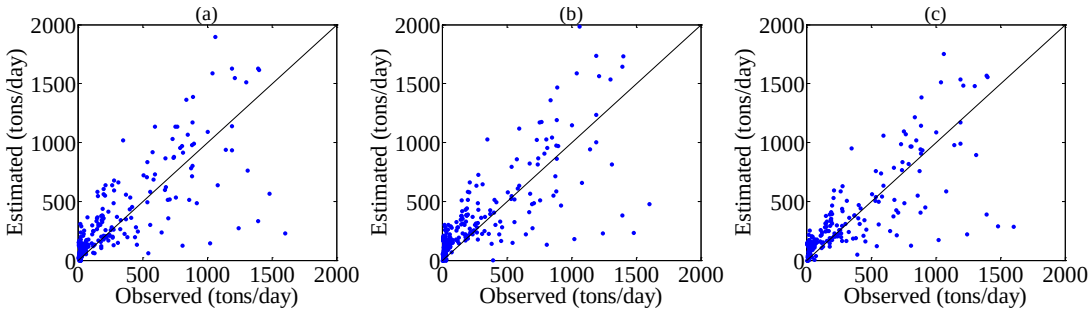

Figure 5.19 Scatter plot of observed and estimated sediment from input combinations (a) 3-1, (b) 3-2 and (c) 3-4 on the Muddy Creek

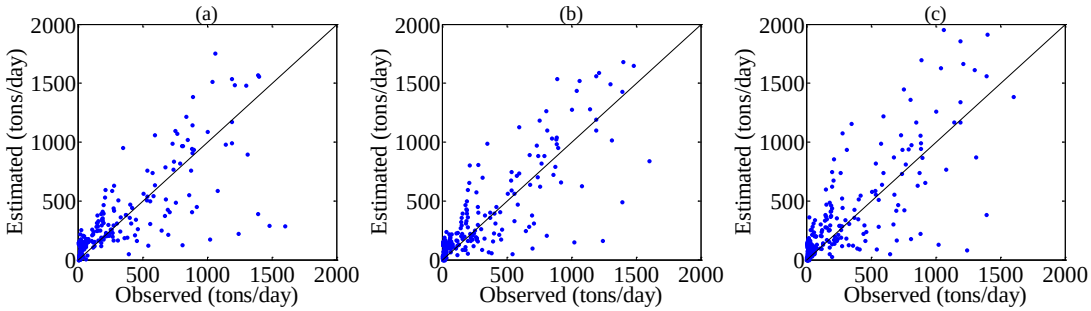

Figure 5.20 Scatter plot of observed and estimated sediment from input combinations (a) 3-4, (b) 3-5 and (c) 3-6 on the Muddy Creek

Likewise, special parameters of the HDFNN model are concerned in this section. The values of four parameters, namely a_0, b_0, recession coefficient λ and initial sediment

deposition SD_0^s for the one-day-ahead estimation are plotted in Figure 5.21. The parameters a_0, b_0 represent a relation between river flow and sediment, and thus vary greatly with the input combinations. Their optimal values are respectively 0.0574 and 5.2456 with the best input combination 1-6. Representative values of 0.6504 and 0.0091 for the parameters λ and SD_0^s are attained as well.

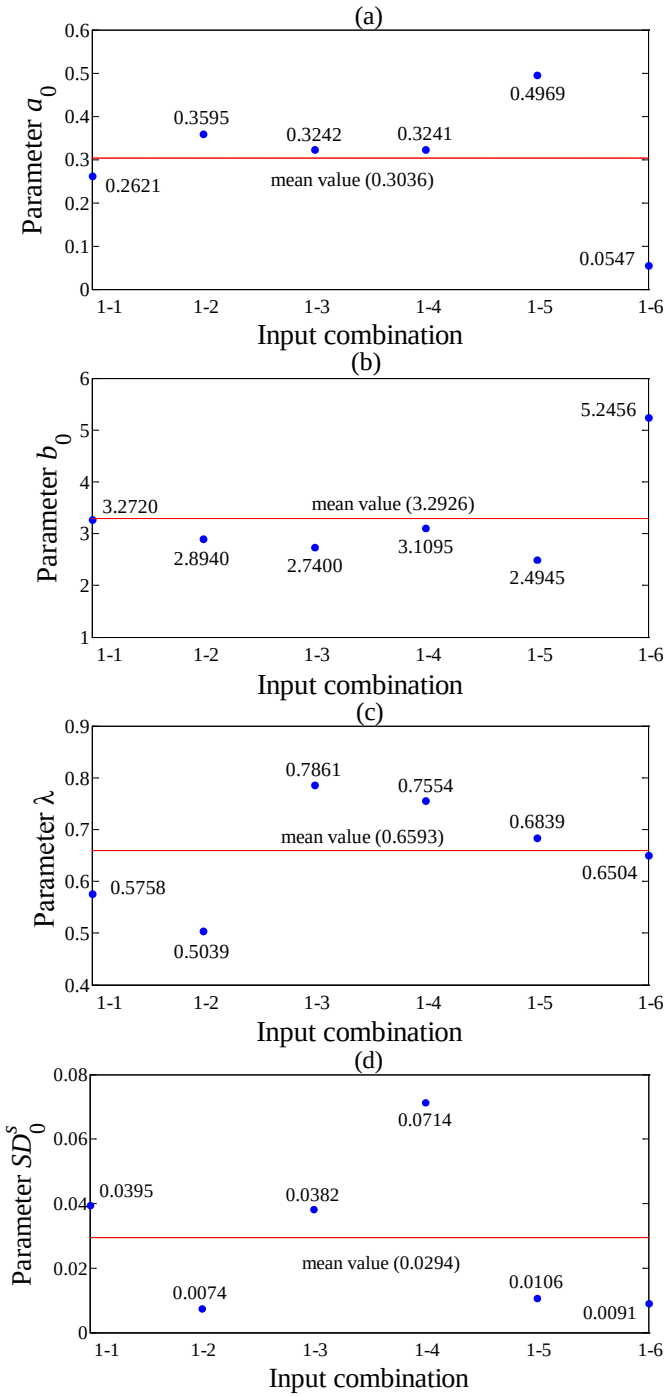

Figure 5.21 Parameter (a) a_0, (b) b_0, (c) λ and (d) SD_0^s computed from different input combinations on the Muddy Creek (one-day-ahead)

The tendency of model performances to input combinations appears to be another matter in the case on the Tongue River. Results of the one-day-ahead estimation in terms of RMSE and NSEC are summarized in Table 5.15. Input combination 1-3 containing $Q_{u(t\text{-}1)}$, $Q_{u(t)}$, $Q_{d(t)}$ and $SSL_{u(t\text{-}1)}$ yields the best results where the NSEC value is as high as 0.9032. The addition of $Q_{d(t\text{-}1)}$ in combination 1-6 generates fairly matched low and medium loads as indicated by the partitioning analysis. The improvements of combination 1-3 over 1-1 and 1-2 are respectively 7.82% and 2.93% in terms of NSEC value while that of 1-6 over 1-4 and 1-5 are respectively 8.12% and 4.17%. The approximated values corresponding to the improvements reveal stable influences of upstream flows on the output. As indicated in Table 5.16, the input variable $Q_{d(t\text{-}2)}$ exerts its effect on two-day-ahead estimation. The input combination 2-5 outperforms 2-3 with a reduction of 12.90% in RMSE value. The latter input combination achieves the best RMSE value in low loads however, revealing the efficiency of input [$Q_{u(t\text{-}1)}$, $Q_{u(t)}$, $Q_{d(t)}$, $Q_{d(t\text{-}1)}$, $SSL_{u(t\text{-}2)}$] in generating low loads. The input combination 2-5 could not effectively fit the medium loads either, in which the RMSE value is 21.45% higher than the 2-4. This difference lies in the inclusion of $Q_{u(t\text{-}1)}$ in combination 2-5. Besides, the addition of $Q_{u(t\text{-}2)}$ in combination 2-6 leads to a 1.96% lower NSEC value when compared with the 2-5. This means that the input $Q_{u(t\text{-}1)}$ helps to match the low and high loads while the $Q_{u(t\text{-}2)}$ seems to be a redundant input variable. Regarding the three-day-ahead estimation in Table 5.17, the input $Q_{u(t\text{-}2)}$ effectively functions on the output since the combination 3-3 renders the best model performances. The inclusion of input variables $Q_{d(t\text{-}3)}$ and $Q_{u(t\text{-}3)}$ in combinations 3-4, 3-5 and 3-6 could not ensure excellent estimations as before. Furthermore, the best performances on low, medium and high loads are obtained from different input combinations. This is another evidence of the uncertainty of the mapping between input and output with the increase of lead time.

Table 5.15 Model performances from different input combinations for the study case on the Tongue River (one-day-ahead)

Input combination	Total RMSE (tons/day)	NSEC	Low RMSE (tons/day)	Medium RMSE (tons/day)	High RMSE (tons/day)
1-1	497.4973	0.8377	222.9459	249.6212	1377.5
1-2	432.1941	0.8775	230.6391	189.0678	1182.2
1-3	**384.1510**	**0.9032**	197.2603	198.1040	**1038.5**
1-4	507.7745	0.8310	198.5880	235.4965	1439.4
1-5	457.9967	0.8625	206.3545	184.6659	1294.4
1-6	393.5422	0.8985	**174.0497**	**178.9218**	1103.3

Table 5.16 Model performances from different input combinations for the study case on the Tongue River (two-day-ahead)

Input combination	Total RMSE (tons/day)	NSEC	Low RMSE (tons/day)	Medium RMSE (tons/day)	High RMSE (tons/day)
2-1	411.0440	0.8895	169.2445	202.7216	1150.2
2-2	431.4866	0.8782	117.5789	160.5999	1271.3
2-3	447.7933	0.8689	**111.0080**	170.3950	1322.3
2-4	411.9271	0.8890	127.4412	**158.7089**	1203.8
2-5	**390.0215**	**0.9005**	125.7231	192.7459	**1113.2**
2-6	423.3997	0.8828	139.5266	183.9982	1221.9

Table 5.17 Model performances from different input combinations for the study case on the Tongue River (three-day-ahead)

Input combination	Total RMSE (tons/day)	NSEC	Low RMSE (tons/day)	Medium RMSE (tons/day)	High RMSE (tons/day)
3-1	449.4354	0.8682	**159.3906**	203.2677	1284.135
3-2	449.2980	0.8683	162.0146	202.8875	1282.307
3-3	**414.9479**	**0.8877**	177.9250	226.0627	**1139.601**
3-4	434.0001	0.8771	191.1288	**202.4367**	1210.981
3-5	457.6289	0.8634	220.9050	228.0081	1251.886
3-6	457.0090	0.8638	179.6844	239.0122	1273.907

The scatter plots of observed and estimated sediments from different input combinations are demonstrated as well, in which the sediments below 500 tons/day and within the range between 2000 and 10000 tons/day are respectively presented to distinguish the model performances on different loads. As shown in Figure 5.22, the impacts of upstream flow are basically embodied in high loads where the distributions of dots computed from combination 1-3 are visibly closer than the others. The three

combinations over-estimate most observations lower than 500 tons/day and the advantage of combination 1-3 over its counterparts is not marked. As for the comparison of downstream flow, Figure 5.23 is plotted with a fix upstream flow input [$Q_{u(t)}$, $Q_{u(t-1)}$]. The addition of $Q_{d(t-1)}$ in combination 2-3 generates closer distributed dots when compared with the 2-1. The superiority of considering $Q_{d(t-2)}$ seems to be revealed from the right figures merely, in which both the combinations 2-3 and 2-5 fairly fit the observations below 500 tons/day. As observed in Figure 5.24, the $Q_{d(t-2)}$ deliver slight differences between combinations 3-1 and 3-2, whereas the inclusion of $Q_{d(t-3)}$ in combinations 3-4 increases accuracy in fitting high loads. Thus a conclusion may be drawn that the downstream flow mainly functions on high loads instead of all values. Model performances in terms of different upstream flows are further shown in Figure 5.25, as a supplement to the single-step-ahead case in Figure 5.23. It can be observed that the combination 3-4 outperforms 3-5 in view of slightly closer distributed dots corresponding to all observations. Thus the $Q_{u(t-2)}$ is proven as a redundant input variable in this case. The addition of $Q_{u(t-3)}$ in combination 3-6 seems not to deteriorate the model performances in comparison with combination 3-5.

The values of a_0, b_0, λ and SD_0^s computed from different input combinations for the one-day-ahead estimation are plotted in Figure 5.26. Their values are respectively recommended as 0.2284, 5.0880, 0.6351 and 0.1221 associated with the combination 1-3. The value of recession coefficient computed from the best input combination for the present case is still smaller than that for the Muddy Creek case, which is highly consistent with the results in Chapter 4. Thus the storage capacity tends to be larger at station on the Muddy Creek than that on the Tongue River.

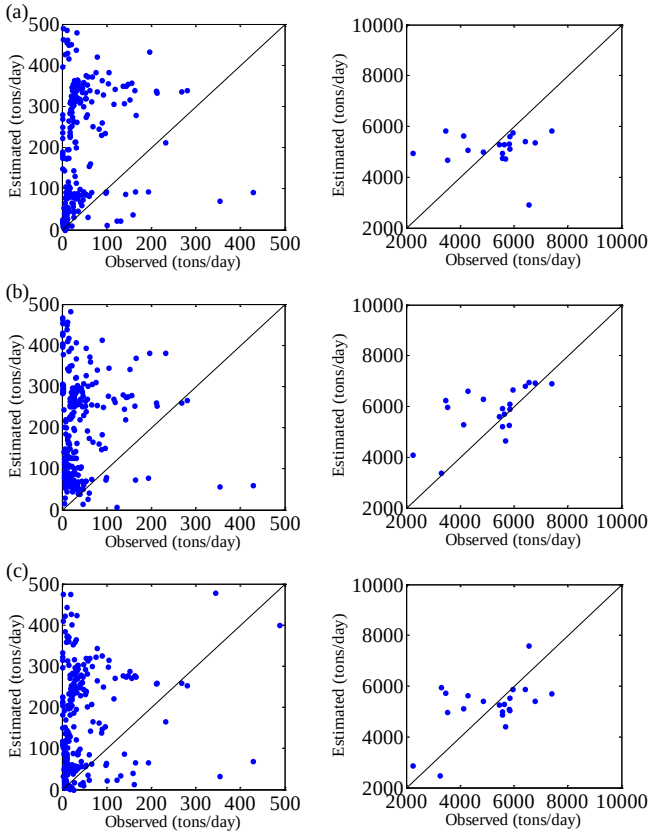

Figure 5.22 Scatter plots of observed and estimated sediment from input combinations (a) 1-1, (b) 1-2 and (c) 1-3 on the Tongue River

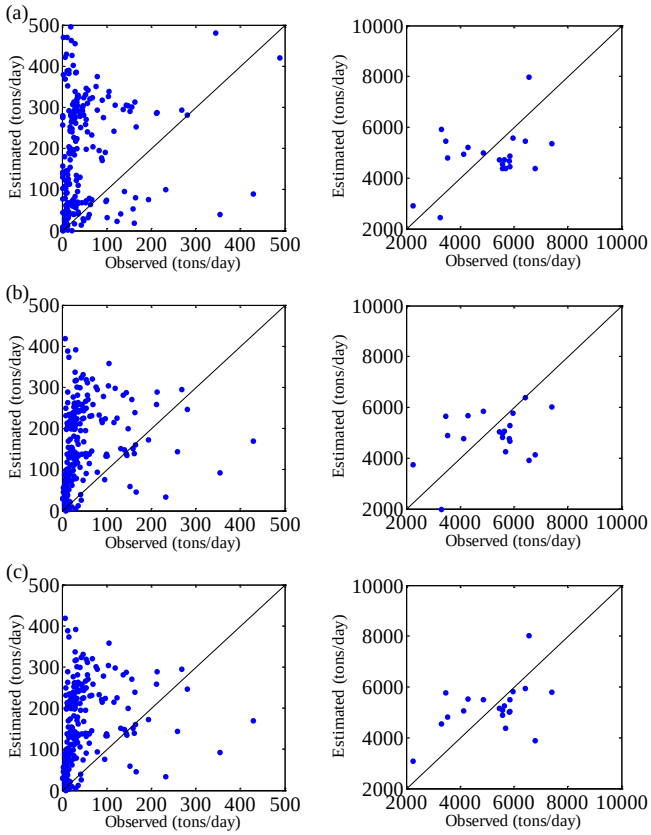

Figure 5.23 Scatter plots of observed and estimated sediment from input combinations (a) 2-1, (b) 2-3 and (c) 2-5 on the Tongue River

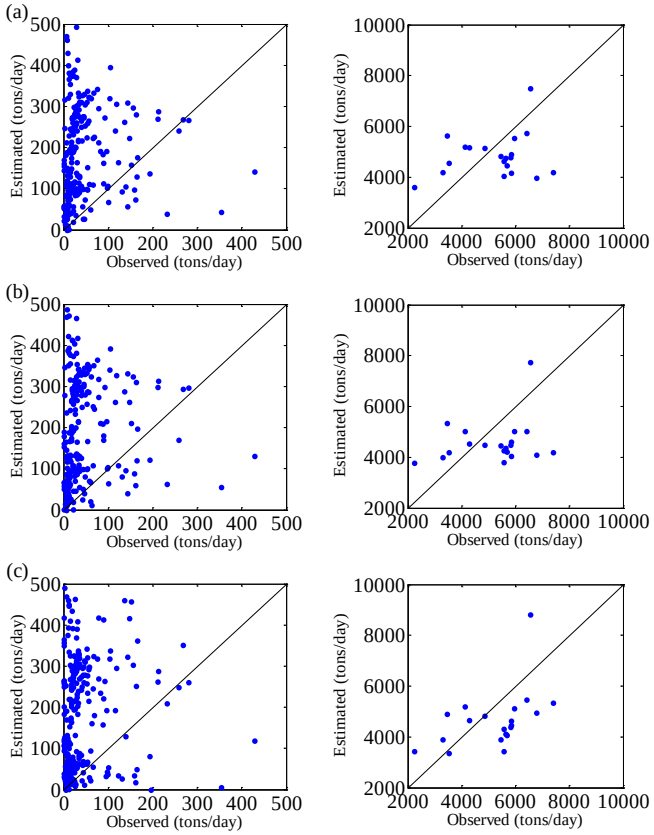

Figure 5.24 Scatter plots of observed and estimated sediment from input combinations (a) 3-1, (b) 3-2 and (c) 3-4 on the Tongue River

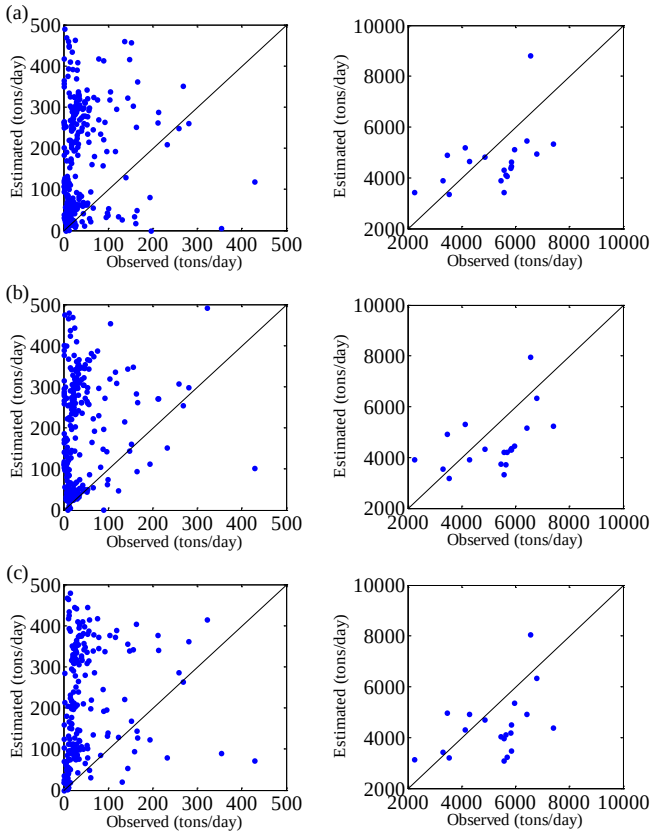

Figure 5.25 Scatter plots of observed and estimated sediment from input combinations (a) 3-4, (b) 3-5 and (c) 3-6 on the Tongue River

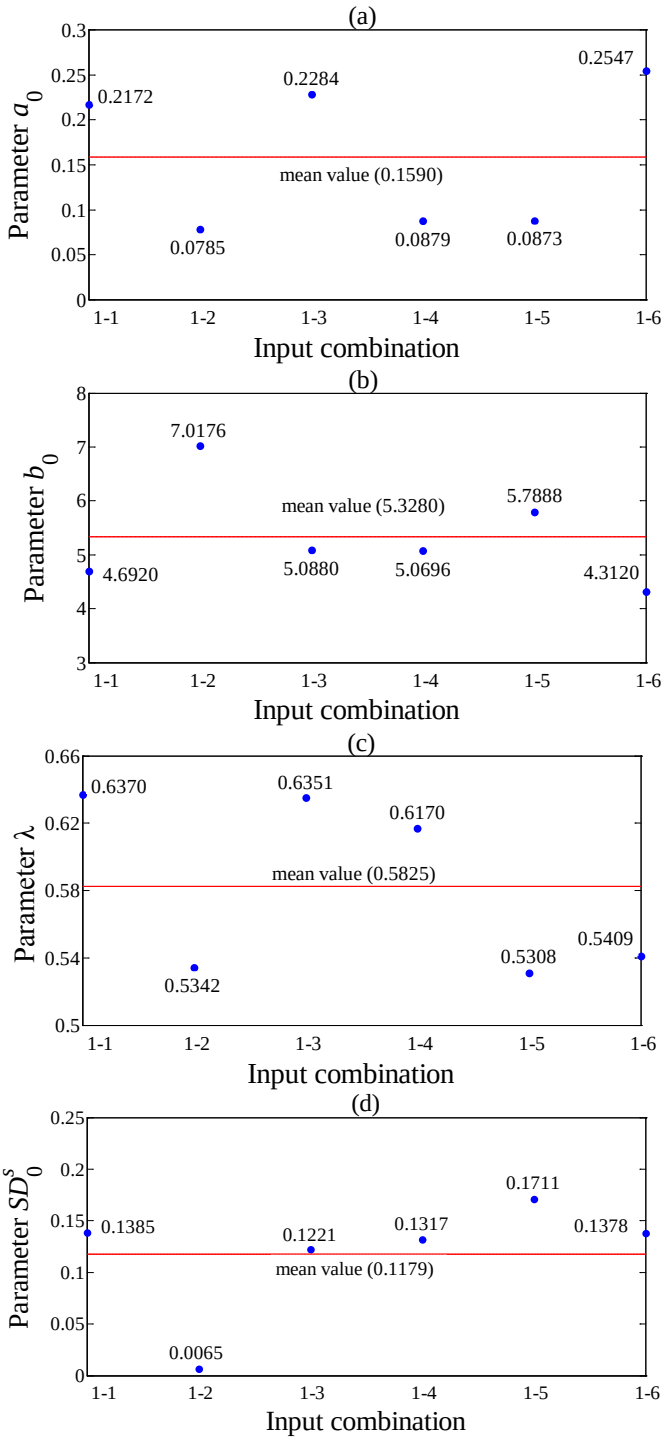

Figure 5.26 Parameter (a) a_0, (b) b_0, (c) λ and (d) SD_0^s computed from different input combinations on the Tongue River (one-day-ahead)

5.4 Summary

The request of determining best input variables for the HNN and HDFNN models has been processed in this chapter. River flow and precipitation data from the upstream sections are imposed as critical inputs for downstream river flow. In particular, areal precipitation over the entire river basin is computed by Thiessen polygon method and regarded as a potential input variable. The Yellow River with four recorded stations in the Altamaha river basin and the North Buffalo creek with three stations within the central Greensboro area are selected as case study sites in the context of daily and hourly forecasts. As far as the sediment load estimation is concerned, river flows from the upstream and downstream stations with different ahead of times are selected to formulate possible input combinations. Stations on the Muddy Creek and the Tongue River are employed as two study cases. The main findings could be highlighted in the following.

(a) The efficiency of flows and precipitations in upstream river sections depends on the basin boundary considered. Areal precipitation is an appropriate input variable coupled with all observed upstream flows in general. The upstream flow inputs may be redundant and uncertain with increasing lead time.

(b) The influences of upstream flow inputs on downstream flow seem to be insensitive if the studied river basin is small. Also due to the shorter distance between upstream and downstream stations, areal precipitation may not be a contributing input variable since the rainfall is not fully collected in time.

(c) Observations on the hourly forecasts reveal that certain input variables including the precipitation may be uncertain and unstable for the HNN model when forecast lead time is shorter than the travel time. The recession coefficients computed from hourly

forecasts are generally larger than that from daily forecasts, implying that the reservoir plays larger storage capacity from the time dimension of hour.

(d) The downstream sediment load seems to be more sensitive to upstream flows than the downstream flows when the studied area is smaller. The downstream flow is identified as a critical factor affecting high sediment loads substantially.

(e) With the increase of lead time, the best performances on low, medium and high loads tend to be attained from different input combinations, revealing the uncertainty of the mapping between input and output.

This chapter provides applicable inputs for downstream river flow and suspended sediment load, particularly putting great concern on the spatial factors. The analysis of input influences is not thorough since, in fact, it is frequently encountered by the difficulty of data measurement and scarcity of relevant data. Besides, mapping between input variables and model output tends to be uncertain with increasing lead time. Data preprocessing methods could be undertaken to eradicate the problem. There is still considerable room for the improvement of the HNN and HDFNN models in order to have excellent overall performances.

6 Uncertainty Analysis on Hydrological Models

In the previous chapters, novel neural network based models were proposed for deterministic predictions on river flows and sediment loads. Uncertainty analysis is complemented in this chapter in order to enhance the reliability and applicability of the models. It is undertaken by incorporating uncertainty into deterministic predictions, and termed interval predictions corresponding to point predictions. By the method of Lower Upper Bound Estimation (LUBE), the suitability of HNN and HDFNN models in producing prediction intervals (PIs) is tested in the following.

6.1 Introduction

Although the applications of data-driven models for the real-time predictions of river flows and sediment loads have been well established over the past years, the analysis of uncertainty is usually disregarded. Uncertainty has always been inherent in hydrological models, which may be imported by inputs, model parameters and structure itself. Generally, the lack of uncertainty analysis can lead to an inflated estimation of the model reliability and applicability, particularly for a novel proposed model. Additionally, in some cases rather than purely the point forecasts, the interval forecasts in which uncertainties are taken into account have become an increasing demand. For these reasons, it is desirable to develop an effective method to deal with the uncertainties in hydrological models.

The quantification of uncertainty in hydrological modeling has attracted considerable attentions in recent years (Mantovan and Todini, 2006; Montanari, 2007; Zou et al.,

2002). The methods to estimate the uncertainty of the model output range from probabilistic forecasting method and sampling-based method, to method based on the analysis of model errors and method based on fuzzy theory (Shrestha and Solomatine, 2006). The probabilistic forecasting method predicts the output by representing the model parameters through probability distributions defined on Bayesian theory, and thereby requiring priori distributions (Krzysztofowicz, 1999). The sampling-based method is generally known as ensemble method or the Monte Carlo method, still in the limit of entailing defining a distribution of parameters (Kuczera and Parent, 1998). The method based on the analysis of model errors analyzes the statistical properties of the errors on account of assumptions for the model residuals (Montanari and Grossi, 2008). The method based on fuzzy theory can be very subjective due to the requisite knowledge of the membership function (Maskey et al., 2004). Alvisi and Franchini (2011) proposed fuzzy neural networks to product crisp forecasts and uncertainty bands and Alvisi et al. (2012) further improved the method by cooperating grey number theory. It was indicated by Jacquin and Shamseldin (2007) that a method based on fuzzy theory is preferred to probability theory when the uncertainties are caused by imprecise knowledge about a real system but not from randomness.

These four methods share a common weakness since they estimate the model uncertainty on specific assumptions. To overcome this issue, an approach based on the model error was improved by Shrestha and Solomatine (2006), which is referred to as an “uncertainty estimation based on local error and clustering”(UNEEC). It takes the joint contribution of all sources of errors into account, drawn on the basis of a clustering approach. The use of the UNEEC method has found increasing applications in the uncertainty assessment since it is not imperative to make any assumption about residual (Dogulu et al., 2014; Pianosi et al., 2010; Solomatine and Shrestha, 2009). Other efforts

are made on the premise of straight-forward and effective methods, which has led to the development of the “Lower Upper Bound estimation” (LUBE). The LUBE method proposed by Khosravi et al. (2011c) constructed the ANN with two output neurons that directly approximated the lower and upper bounds. It made no assumption about the data distribution. The LUBE method has proven to be simple, fast and efficient in generating high quality PIs (Khosravi et al., 2011c; Quan et al., 2014a, 2014b). Taormina and Chau (2015a) successfully combined the LUBE with Multi-Objective Fully-Informed PSO for neural network river forecasting. Besides, the LUBE method has been applied to many other fields, such as travel time prediction (Khosravi et al., 2011a), electricity load prediction (Rana et al., 2013) and flood forecast (Ye et al., 2016).

The main contribution of this chapter is to apply the LUBE method to neural network based models for PIs construction. The rest of the chapter is organized as follows. Section 6.2 introduces the LUBE method with neural network based models and the indices used for PIs evaluation. Cases studies are performed in Section 6.3 and 6.4 respectively for the river flow forecasting and sediment load estimation. The suitability of proposed HNN and HDFNN models is examined by assessing the qualities of PIs generated, in comparison with their corresponding benchmark models. Section 6.5 finally concludes the results and discussions.

6.2 Estimation of neural network based PIs with LUBE

By definition, a PI is a range of values with lower and upper bounds for a prescribed coverage probability called the confidence level (CL). As such, the PI is constructed by predicting the lower and upper bounds and has an indication of accuracy with

confidence level. Suppose (1-*a*)% is the target CL for output variable *Y*, *l* and *u* are lower and upper bounds in which PI is delimited as [*l*, *u*] when Pr (*l*<*Y*<*u*)=1-*a*. That is, each value of the predicted output is enclosed in the intervals with (1-*a*)% probability. For a given confidence level, narrower width of the intervals indicates less uncertainty of the prediction and accordingly entails higher accuracy. Coverage probability and width of intervals are two important indices for evaluation of PIs quality. They are represented by Prediction Interval Coverage Probability (PICP) and Prediction Interval Normalized Root-mean-square Width (PINRW) in literatures. A compound PI-based indicator, namely Coverage Width-based Criterion (CWC), is employed in the training period as an objective function. The definitions of these indices are presented as follows.

The PICP measures the percentage of target observations that are enclosed in the intervals (Khosravi et al., 2010):

$$\mathrm{PICP} = \frac{1}{n}\sum_{i=1}^{n} c_i \tag{6.1}$$

where n is the number of observations, c_i is equal to one if the observation y_i lies within the range of $[l_i, u_i]$ and zero otherwise. It is obvious that when all observations are covered by intervals, PICP=100% as the ideal case. The PINRW is defined as the ratio of the 2-norm of the width of the PIs to the range R of the output variable (Quan et al., 2014a):

$$\mathrm{PINRW} = \frac{1}{R}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(u_i - l_i)^2} \tag{6.2}$$

It is inspired by the successful applications of RMSE for the point predictions. The 1-norm of the width of the PIs, namely Prediction Interval Normalized Average Width (PINAW), can also be employed instead of PINRW. It is defined as the ratio of the

average width of the PIs to the range *R* of the output variable (Khosravi et al., 2011b):

$$\text{PINAW} = \frac{1}{nR}\sum_{i=1}^{n}(u_i - l_i) \tag{6.3}$$

In this study, PINRW is used during the training period since it magnifies wider intervals and trends to obtain better results. The PINAW is employed for the testing for an overall evaluation of the width of PIs. It can be seen that larger PICP and narrower PINRW indicate higher quality PIs. However, they are two conflicting properties. For the purpose of comprehensive balance between the PICP and PINRW, a cost function CWC is developed as follows (Quan et al., 2014a):

$$\text{CWC} = \text{PINRW}(1 + \gamma(\text{PICP})e^{-\eta(\text{PICP}-\mu)}) \tag{6.4}$$

where $\gamma(\text{PICP}) = 1$ for training in order to construct more conservative PIs. It becomes a step function of PICP in the testing period:

$$\gamma(\text{PICP}) = \begin{cases} 0 & \text{if PICP} \geq \mu; \\ 1 & \text{if PICP} < \mu. \end{cases} \tag{6.5}$$

The parameters η and μ are two constants that determine the penalty term controlling the balance between coverage probability and interval width. The value of μ can be set to (1-*a*) since it is the nominal confidence level associated with PIs. η exponentially magnifies the difference between PICP and μ, and is suggested to be valued as 80 (Quan et al., 2014a). CWC tries to find a trade off between valid PIs for which PICP≥(1-*a*) holds and informative PIs with narrow width in the optimization process.

Lower Upper Bound Estimation (LUBE) method is a straightforward technique to generate the lower and upper bounds as two outputs for neural network based models. As demonstrated in Figure 6.1, the HNN model directly obtains two outputs $Q^{out}_{u(T+\Delta T)}$ and $Q^{out}_{l(T+\Delta T)}$, corresponding to the upper and lower bounds of the PIs separately. Accordingly, two storage reservoirs with different recession coefficient and initial

storage are assumed as two downstream sections and continuity equation is still satisfied when river flows from the upstream section to the downstream ones. The set of parameters to be adjusted is $\{w_{11},...,w_{ks},\overline{w}_{11},...,\overline{w}_{s1},\overline{w}_{12},...,\overline{w}_{s2},\overline{\lambda}_1,\overline{\lambda}_2,\overline{S}_{1(T=0)},\overline{S}_{2(T=0)}\}$ correspondingly, in which weights $\{w_{11},...,w_{ks}\}$ from the input to the hidden layer are used for both output of the HNN-LUBE model. In other words, the PI constructed by LUBE method is the range of values delimited by the two outputs produced by the HNN model. The flowchart in Figure 6.2 demonstrates how HDFNN-LUBE is implemented with reference to the topological structure of HDFNN model presented in Chapter 4. The upper and lower bounds of the PIs are directly computed with two outputs SSL_u and SSL_l, in which the structure of double feedforward neural network is still maintained. The values of bounds SSL_u and SSL_l respectively contain two parts of results, which are obtained from discharge and sediment inputs separately. It is supposed that observations fall between the two HDFNN outputs with (1-a)% probability. Unlike traditional methods for construction of PIs, the LUBE is a simple method of release from estimating the mean and variance values of point prediction. It can be combined with neural network based models to directly generate the upper and lower bounds of PIs in one step.

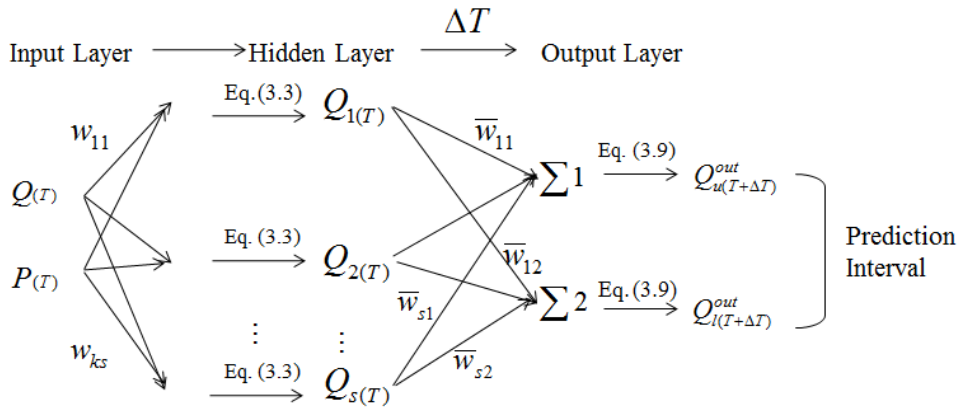

Figure 6.1 HNN model for LUBE to generate upper and lower bounds of PIs

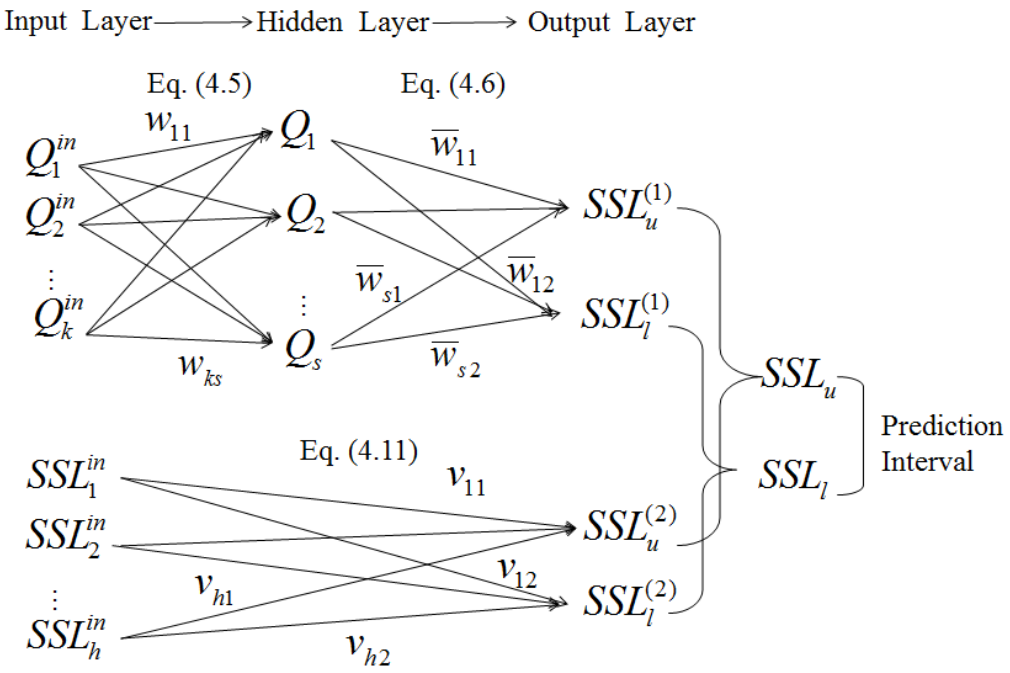

Figure 6.2 HDFNN model for LUBE to generate upper and lower bounds of PIs

The training of neural network based model for LUBE method is performed by minimizing the cost function CWC, which is nonlinear, discontinuous and non-differentiable. Traditional derivative-based optimization algorithm could not be employed for its minimization. In this chapter, differential evolution (DE) is applied as an optimization algorithm to minimize the cost function during the training period. To facilitate the model training, the entire data are scaled linearly to the range between 0 and 1. They are then separated into training, validation and testing sets. During the calibration process, the training is strictly terminated at the point where the error in the validation set begins to rise in order to avoid over-fitting the training data. The testing data are used to evaluate the quality of PIs, which would be measured by four indices namely PICP, CWC, PINRW and PINAW expressed in terms of percentages. In the following, the LUBE method is employed for construction of PIs and to quantify uncertainties of the HNN and HDFNN models respectively for downstream river flows and suspended sediment loads.

6.3 Uncertainty analysis for river flow forecasting

The LUBE method for construction of PIs is applied in the river flow forecasting models in the present section. The main objective is to assess the reliability of HNN model proposed in Chapter 3, by quantifying the uncertainties in comparison with its counterpart models namely ANN, FPNN and HYMN. The Yellow River in the Altamaha river basin is employed as a case study site, in which river flow at the downstream station 02208000 is the target output. Both single- and multi-step-ahead cases are performed for a comprehensive analysis in the following.

6.3.1 Performances on single-step-ahead forecasting

As a study case, single-step-ahead forecasting in the Altamaha river basin is presented in daily scale. The PIs are developed using input variables of three upstream flows and areal precipitation. This corresponds to the input combination 3-3 in Chapter 5, which is the optimal input for one-day-ahead point forecasting. Uncertainty analysis is undertaken with CL of 90%, 95% and 99% in conjunction with LUBE method, thus the value of μ in Eq. (6.4) is set to 0.90, 0.95 and 0.99, respectively. To begin with, the optimal number of nodes in the hidden layer is determined by gradually increasing the node number and finding the best results for each model. The procedure is not presented in details herein since similar practice has been demonstrated in Chapter 3. The results are not the same with the point predictions, however. Taking the case with 90% confidence level as an example, the optimal numbers of nodes in the hidden layer are 8, 10, 9 and 11 respectively for the ANN, FPNN, HYMN and HNN models.

The comparisons of the 90%, 95% and 99% PIs by various models in terms of PICP, CWC, PINRW and PINAW are summarized in Table 6.1. Taken as a whole, the values

of CWC, PINRW and PINAW become larger with increasing confidence level and PICP value. It is reasonable since a narrow width of intervals could not be guaranteed if the intervals intend to enclose many observations. With respect to the 90% CL, the PICP values attained by the four models are larger than 0.90, indicating a satisfactory coverage probability of the PIs. On this basis, CWC retrogrades to be the same as the PINRW index considering the interval width only. Herein, PINRW and PINAW are respectively 2-norm and 1-norm of the width of the PIs for evaluation. It can be observed from the table that the HNN model yields the smallest PINRW and PINAW values among the four models. There are respectively 18.64%, 11.33% and 8.82% decrements in terms of PINRW when compared with ANN, FPNN and HYMN models. The suitability of HNN-LUBE model in producing PIs is thus ensured at 90% confidence level.

As for the confidence level 95%, the four forecasting models still return strictly valid PIs with corresponding PICP values of 95.06%, 97.25%, 95.33% and 95.60%. With almost unqualified coverage probability and relatively wide intervals of PIs, the ANN model might be not efficient to deal with the uncertainties of the river flows at 95% CL. The FPNN obtains larger PICP and smaller PINAW values in comparison with the ANN, and yet their PINRW values are nearly equal. This means that the FPNN model is available to enclose more observations than the ANN with nearly the same interval width. On the contrary, the HYMN contains almost the same amount of observations with narrower intervals when compared with the ANN model. The qualities of PIs are improved by the FPNN and HYMN model therefore, where higher accuracies are achieved with less uncertainty of the forecast. By integrating the PFNN and HYMN models, the HNN could better describe the complex features of river flows and thus demonstrate applicability in the forecasting. It is verified by the smallest PINRW and

PINAW values obtained, in which the reduction of PINAW is respectively 32.37%, 23.43% and 6.64% compared with the other three models. Likewise, all the four models generate valid PIs for the 99% case and the best performer with the highest quality PIs is obtained by the HNN model. It seems that the improvement of HNN model over the others is not as significant as in the cases of 90% and 95% CL. The probable reason is that the parameters are not easily to be searched to fit both the lower and upper bounds of the PIs with high confidence level. As a whole, the four forecasting models optimized by the DE algorithm are capable of achieving convincing PIs using LUBE for all the examined cases. The HNN model unsurprisingly returns the narrowest PIs, with PINAW of 15.31%, 16.44% and 32.74% for the one-day-ahead forecasting. It is interesting to note that the PICP values obtained by the HNN model are not significantly higher than the corresponding CL values, but the CWC values are quite small. This provides further evidence for the reliability of HNN model, in that it could construct narrow PIs with highly efficient coverage probability.

The LUBE generated PIs at 95% confidence level for the one-day-ahead river flow forecasting is demonstrated in Figure 6.3. It is noticed that the ANN presents better performances for the first 200 around observations than the FPNN model, since the latter generates apparently inferior lower bounds which are far from the observations. On the contrary, the PIs constructions of upper bounds are more appreciated by the FPNN model. Besides, the ANN could not generate excellent PIs for the latter part of observations, as implied by the almost out-of-bounds of observations and wide intervals. From a comparison between Figure 6.3(b) and (c), it emerges that the HYMN model generally attains closer upper and lower bounds of PIs, although failing in simulating appropriate upper bounds of some high values. It can be detected in Figure 6.3(d) that the PIs constructed by HNN model are evidently narrower than those by the other three,

which conveys more information with less uncertainty. The superiority of HNN over FPNN model is reflected into a better positioning of lower bounds of the PIs, where the continuity equation integrated in HNN model facilitates the computing particularly in small values. It might be due to that the lower bounds of PIs could be separately adjusted by the recession coefficient of the assumed reservoir, to effectively change with the complexly varied observation series. On the other hand, the HNN is better than HYMN model in terms of the high values of observations. The different patterns classified by fuzzy pattern-recognition in the HNN model might explain the excellent performances in capturing high values. Besides, the abolishment of impractical assumed reservoirs in the hidden layer renders the HNN more compact with narrower PIs than the HYMN model. It is thus concluded that the HNN model could produce convincing upper and lower bounds with high quality PIs for the present one-day-ahead forecasting.

Table 6.1 Testing performances by various models for the one-day-ahead case in the Altamaha river basin

	models	PICP	CWC	PINRW	PINAW
90% confidence level	ANN	0.9231	0.2377	0.2377	0.2006
	FPNN	0.9011	0.2181	0.2181	0.2068
	HYMN	0.9176	0.2121	0.2121	0.1950
	HNN	0.9093	0.1934	0.1934	0.1531
95% confidence level	ANN	0.9506	0.2831	0.2831	0.2431
	FPNN	0.9725	0.2825	0.2825	0.2147
	HYMN	0.9533	0.2256	0.2256	0.1761
	HNN	0.9560	0.2005	0.2005	0.1644
99% confidence level	ANN	0.9918	0.4690	0.4690	0.3902
	FPNN	0.9945	0.4563	0.4563	0.3780
	HYMN	0.9918	0.4269	0.4269	0.3907
	HNN	0.9973	0.3917	0.3917	0.3274

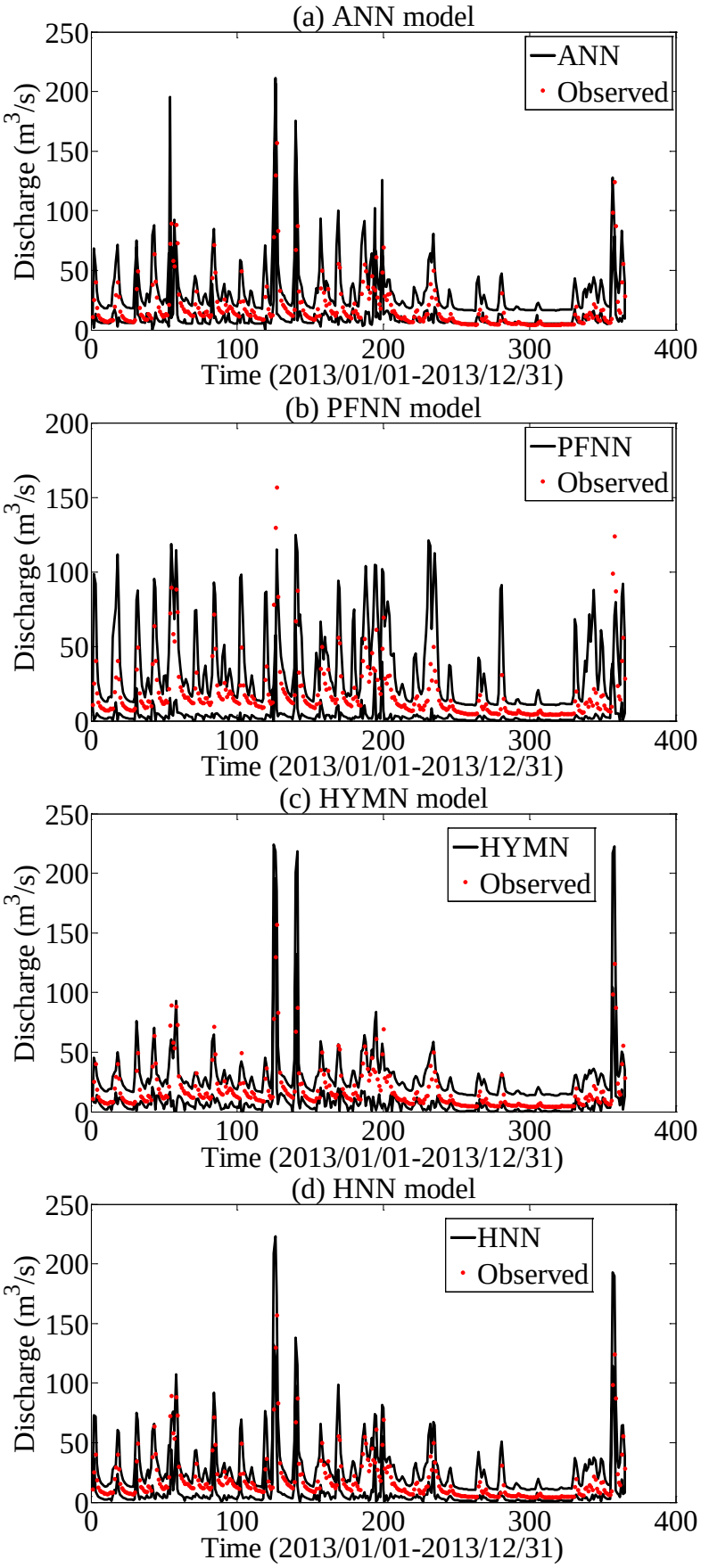

Figure 6.3 LUBE generated PIs at 95% confidence level for the one-day-ahead case in the Altamaha river basin by (a) ANN (b) PFNN (c) HYMN and (d) HNN model

6.3.2 Performances on multi-step-ahead forecasting

As far as the performances of forecasting models on multi-step-ahead uncertainty predictions are concerned, PIs with a lead time of 16 hours are produced for the USGS gauging stations in the Altamaha river basin. It has been found in Chapter 3 that the ANN and FPNN models are not available to perform the 16-hour-ahead point forecasting while the HYMN and HNN models present their efficiency with input variables of the nearest upstream flow and point precipitation. The modeling in this section utilizes the same input variables for the interval prediction, which have proven as the best input in Chapter 5. The applicability and credibility of the four models with large forecast lead time are accordingly assessed in the following.

Details regarding the model performances in PIs constructions are reported in Table 6.2 at 90%, 95% and 99% CL. With respect to the 90% case, the ANN and FPNN models yield PIs enclosing the same number of observations. The width of intervals obtained by the FPNN model is slightly smaller than the ANN. The HYMN and HNN generate narrower intervals than the above two models, with smaller coverage probability of 90.70%, however. The HNN constructs the narrowest PIs with a slight reduction of 2.12%, 1.16% and 1.17% in terms of PINRW when compared with ANN, FPNN and HYMN models. It seems that the PIs generated by the HNN do not have significantly higher quality than the other models. This can probably be attributed to the large forecast lead time, where the parameters of recession coefficient and initial storage might have large discrepancy for the corresponding assumed reservoir. Results with confidence level 95% reveal more improvement of HNN model over the others where the decrements of PINRW are 12.58%, 9.16% and 5.87%, respectively. The corresponding PICP values of 95.93%, 95.93%, 95.35 % and 96.22% for the 99% case imply that the four models are incapable of satisfying the validity condition. These

unqualified PIs are accompanied with large CWC values, which are not equal to the corresponding PINRW values. Notwithstanding, the HNN is the most viable model with largest coverage probability and narrowest intervals. The reliability of forecasting models with large lead time is not always guaranteed if the PIs are proposed to enclose observations with a high coverage probability. This is probably due to the increasing uncertainties accompanied with inputs of large forecasting lead time.

Table 6.2 Testing performances by various models for the 16-hour-ahead case in the Altamaha river basin

	models	PICP	CWC	PINRW	PINAW
90% confidence level	ANN	0.9273	0.3492	0.3492	0.3092
	FPNN	0.9273	0.3458	0.3458	0.3050
	HYMN	0.9070	0.3459	0.3459	0.3022
	HNN	0.9070	0.3418	0.3418	0.2975
95% confidence level	ANN	0.9651	0.4674	0.4674	0.4331
	FPNN	0.9564	0.4498	0.4498	0.4029
	HYMN	0.9535	0.4341	0.4341	0.3927
	HNN	0.9564	0.4086	0.4086	0.3567
99% confidence level	ANN	0.9593	7.1481	0.5648	0.5120
	FPNN	0.9593	7.0320	0.5556	0.5017
	HYMN	0.9535	10.9104	0.5578	0.5002
	HNN	0.9622	5.5637	0.5435	0.4881

In order to visually observe the performances of various models for the 16-hour-ahead forecasting, the LUBE generated PIs at 95% confidence level are plotted in Figure 6.4. As can be seen, there are obvious lagged forecasts of the intervals produced by the four models, which are in advance of the observations. This might be acceptable for forecasting applications, and could be eliminated by reliable data preprocessing techniques. The valid PIs are also found to comprise the peak discharge for the major storm event. As observed in Figure 6.4(a), the PFNN model is superior to the ANN in generating closer lower bounds of PIs to the observations. The high values of the observations are not appropriately simulated, in which the corresponding upper bounds stay a bit away from the observed ones. Herein, the PFNN presents slightly worse

performances than the ANN in the PIs construction of peak values. As for the HYMN and ANN models demonstrated in Figure 6.4(b), the HYMN model produces better lower bounds of PIs than the ANN. Thus the qualities of PIs are improved by the FPNN and HYMN models in light of better lower bounds constructions in comparison with the ANN. Observations of the Figure 6.4(c) and (d) reveal that the superiority of HNN model over the FPNN and HYMN lies in the excellent generation of upper bounds of PIs. The width of PIs by the HNN is therefore, evidently the smallest among the four models. These comparisons conclusively reveal that the HNN model is able to produce more effective upper and lower bounds of the PIs using LUBE than the traditional ANN model. Its applicability is thus validated for the current 16-hour-ahead forecasting, in the light of less uncertainty in the narrow PIs with 95% confidence level.

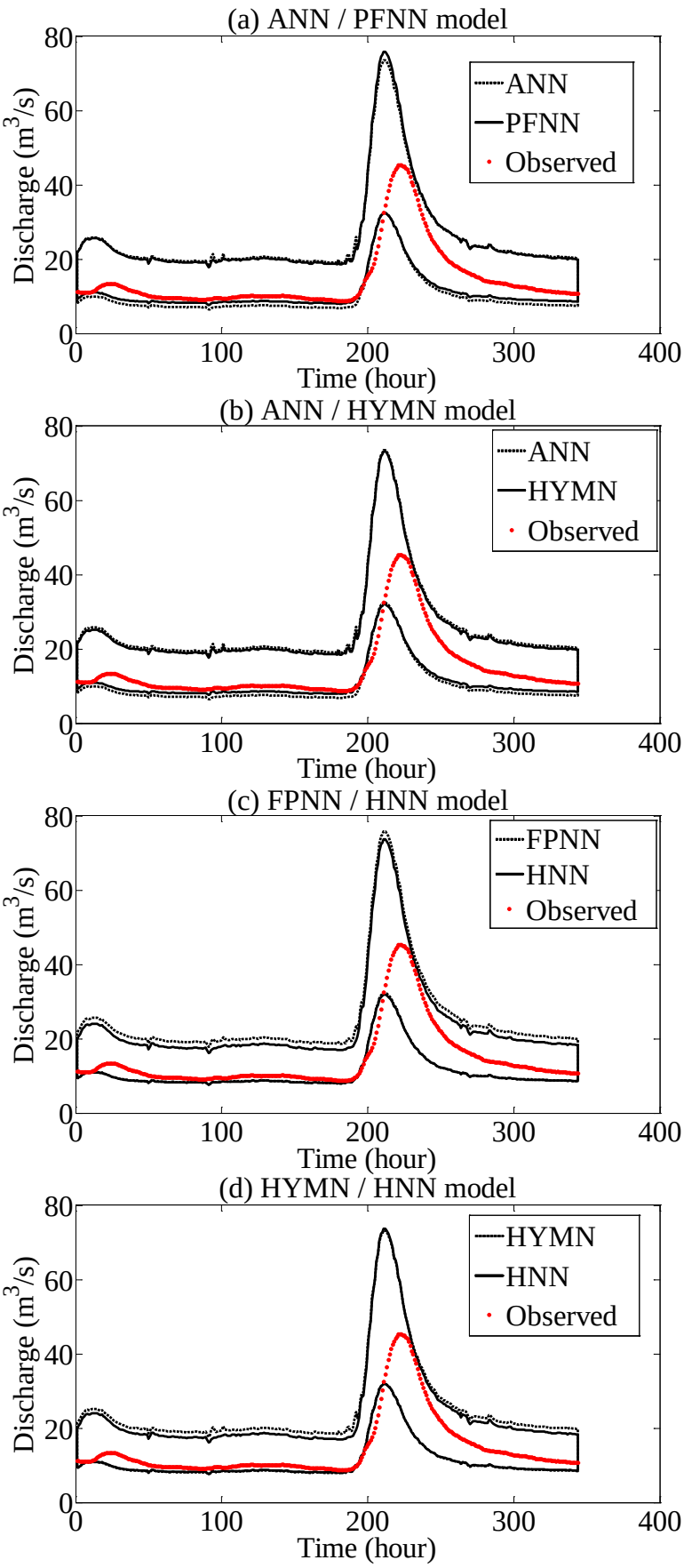

Figure 6.4 LUBE generated PIs at 95% confidence level for the16-hour-ahead case in the Altamaha river basin by (a) ANN/PFNN (b) ANN/HYMN (c) FPNN/HNN and (d) HYMN/HNN model

6.4 Uncertainty analysis for sediment load estimation

The applications of LUBE method on estimation models for suspended sediment load (*SSL*) are presented in the following section. The Muddy Creek in Montana is considered as a study case site, in which the *SSL* at downstream station 06088500 is the output variable. Daily data are applied to models for both single- and multi-step-ahead estimation. The comparative study is undertaken between HDFNN model and its counterpart models namely MFNN, DPFNN and HFNN, to examine their abilities in constructing prediction intervals using LUBE method.

6.4.1 Performances on single-step-ahead estimation

As indicated in Chapter 5, input combination consisting of $Q_{u(t\text{-}1)}$, $Q_{u(t)}$, $Q_{d(t)}$, $Q_{d(t\text{-}1)}$ and $SSL_{u(t\text{-}1)}$ attains the best point prediction for the one-day-ahead case on the Muddy Creek. It is adopted in the current section as well for the uncertainty analysis of the estimation models. In order to search optimal model structure, the number of the nodes in the hidden layer is tested from 2 to 15 for each model employed.

Table 6.3 shows the comparisons of the 90%, 95% and 99% PIs constructed by MFNN, DPFNN, HFNN and HDFNN models. The four models generate valid PIs for the 90% CL, indicated by the PICP values larger than 0.90. Thus the values of CWC are the same with those of PINRW, and employed together with PINAW as a width index to evaluate the quality of PIs. The corresponding values of PINRW and PINAW gradually decrease by the four models, in which the HDFNN generates the narrowest intervals. The decrements of PINAW by the HDFNN are respectively 22.22%, 17.97% and 13.22% when compared with MFNN, DPFNN and HFNN. In regard to the 95% CL, the improvement of HDFNN over the other three models is not that significant, e.g., with a

13.07% reduction in PINAW value in comparison with MFNN model. Despite that, its superiority in construction of PIs is clearly exhibited with the largest PICP and smallest PINRW/PINAW values. For the present one-day-ahead estimation, the four models are available to produce PIs covering more than 99% observations, thus their reliability could be ensured with 99% confidence level. However, the interval widths remarkably increase and the qualities of PIs are not as excellent as those with 90% and 95% confidence levels. For instance, the PIs with 99% CL covers more than 1.98% observations than the 95% case by the HDFNN model, but the corresponding width of intervals is 126.80% larger in terms of PINAW. This might probably be explained by the same reasons with the forecasting models, that the given high confidence level increases difficulty in adapting both the lower and upper bounds of the PIs using the same parameters from the input to the hidden layer.

Table 6.3 Testing performances by various models for the one-day-ahead case on the Muddy Creek

	models	PICP	CWC	PINRW	PINAW
90% confidence level	MFNN	0.9588	0.0172	0.0172	0.0135
	DPFNN	0.9588	0.0158	0.0158	0.0128
	HFNN	0.9533	0.0150	0.0150	0.0121
	HDFNN	0.9506	0.0137	0.0137	0.0105
95% confidence level	MFNN	0.9670	0.0217	0.0217	0.0176
	DPFNN	0.9670	0.0204	0.0204	0.0168
	HFNN	0.9698	0.0197	0.0197	0.0158
	HDFNN	0.9725	0.0182	0.0182	0.0153
99% confidence level	MFNN	0.9918	0.0657	0.0657	0.0450
	DPFNN	0.9945	0.0599	0.0599	0.0443
	HFNN	0.9918	0.0514	0.0514	0.0406
	HDFNN	0.9918	0.0445	0.0445	0.0347

For further study of PIs constructions by various models, partitioning analysis is employed to provide explicit performances on different intervals of values. As suggested in Chapter 4, median and mean of the dataset in the testing period are considered as two threshold values, and the entire data are divided into low, medium and high loads accordingly. The statistics of different loads obtained by the four models

at 95% confidence level are displayed in Table 6.4. As can be seen, the PIs are capable of enclosing all observations located in low and medium loads. The PICP values in high loads are respectively 83.10%, 83.10%, 84.51% and 85.92% for the four models, which are far from the confidence level of 95%. Thus the PIs in terms of high loads are considered completely invalid. The high discrepancy and outlier presence of sediment data render it difficult to effectively capture both the low and high sediment loads, which is also a challenge in the point predictions. As expected, the HDFNN model produces narrow PIs in terms of different loads, which finally lead to the smallest PINRW and PINAW values obtained using the whole dataset in Table 6.3. In regard to the low load, the width of intervals based on PINAW are respectively 17.24%, 14.28% and 6.49% smaller than the MFNN, DFPNN and HFNN models. The HFNN model produces slightly better PIs than the HDFNN in terms of medium loads. This might be attributed to many reasons, for instance, the fuzzy patterns in the HFNN happen to capture the medium observations effectively for PIs construction. However, the performance of HFNN model has no general significance, which could not be concluded that it is more reliable than the HDFNN. Albeit being invalid, the PIs attained by the HDFNN model have the highest quality in high loads. It covers 1.67% observations more than the HFNN model with 8.33% smaller interval width. Accordingly, PIs for the sediment simulation mainly present their efficiency in low and medium loads. The HDFNN model is able to generate relatively excellent PIs in terms of high loads, in spite of unconvincing at a particular confidence level.

Table 6.4 Statistics of different loads by various models for the one-day-ahead case on the Muddy Creek at 95% confidence level

Model	Low load			Medium load			High load		
	PICP	PINRW	PINAW	PICP	PINRW	PINAW	PICP	PINRW	PINAW
MFNN	1	0.0090	0.0087	1	0.0229	0.0217	0.8310	0.0374	0.0346
DPFNN	1	0.0087	0.0084	1	0.0215	0.0199	0.8310	0.0351	0.0399
HFNN	1	0.0082	0.0077	1	0.0196	0.0189	0.8451	0.0351	0.0324
HDFNN	1	0.0079	0.0072	1	0.0201	0.0194	0.8592	0.0303	0.0297

The PIs generated at 95% CL by the four models are depicted in Figure 6.5 for comparison. For convenience of visual inspection, six points of observations with extremely high values are cut from the time series and not demonstrated in the figures. As can be seen from Figure 6.5(a), the MFNN and DFPNN are not available to generate effective lower bounds of PIs for the middle part of the observations. The lower bounds obtained are nearly equal to a small value and are not consistent with observations in the flow series. With respect to the upper bounds of PIs, the performances of DFPNN are slightly superior to the MFNN for the middle part of the observations, nevertheless, comparable with the rest of observations. The more compact structure of DFPNN model might explain the above performances, that less uncertainty might be accompanied with more optimal structure of models. The integration of fuzzy pattern-recognition into the HFNN model render it generate PIs with better upper bounds than the MFNN, as indicated in Figure 6.5(b). The performances presented in Figure 6.5(c) imply that the HDFNN model is particularly outstanding in construction of lower bounds of PIs. This might be explained by the practice of introducing continuity equation in HDFNN model, in which the lower bounds of PIs could be effectively produced using the recession coefficient, and change with the time series of observations. Narrower intervals by HFNN and HDFNN models are also observed when compared with that by the MFNN and DPFNN. The capacity of the HDFNN model in making appropriate uncertainty prediction is thus verified in view of producing suitable lower bounds and narrower intervals.

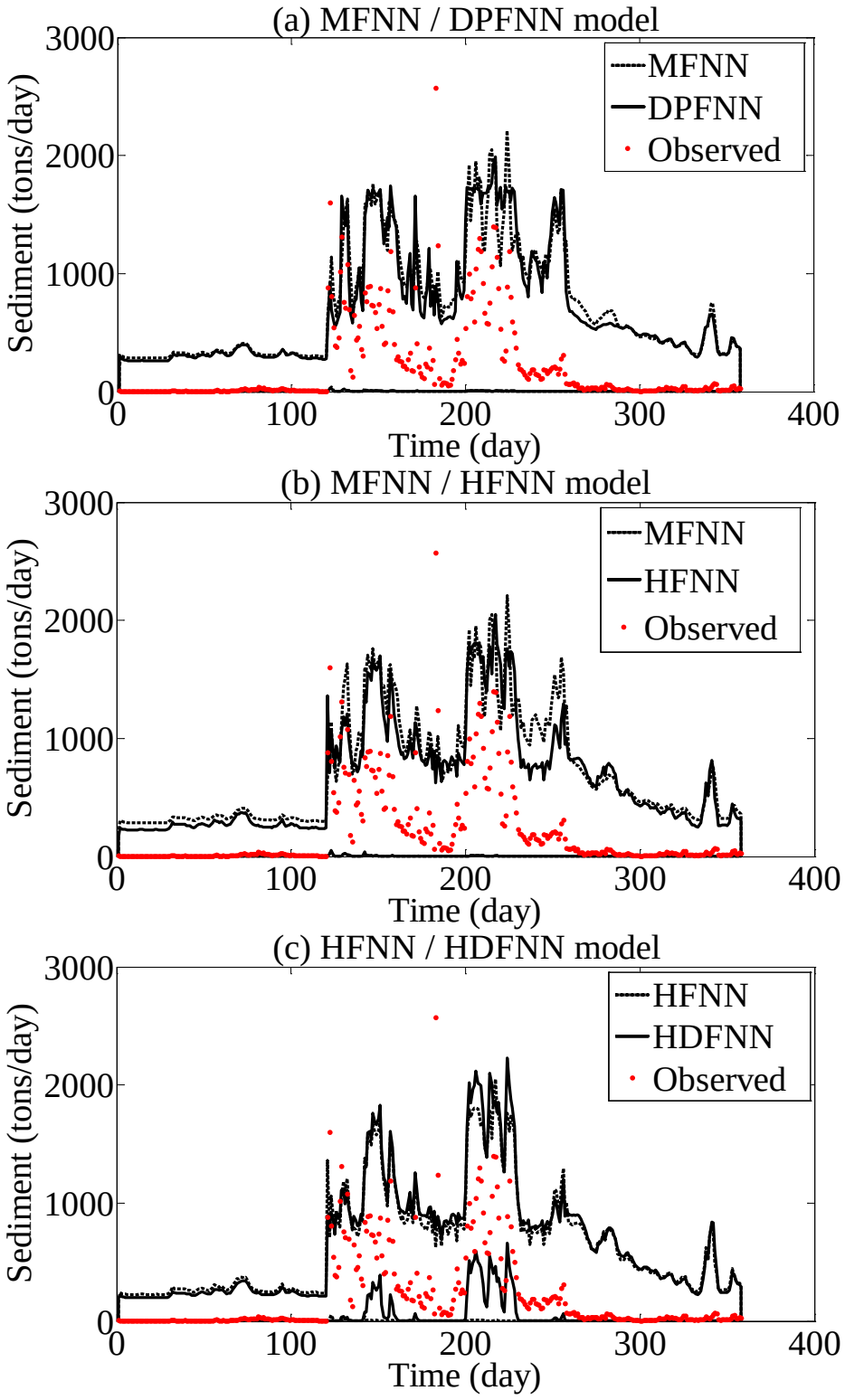

Figure 6.5 LUBE generated PIs at 95% confidence level for one-day-ahead case on the Muddy Creek by (a) MFNN/DPFNN (b) MFNN/HFNN and (c) HFNN/HDFNN models

6.4.2 Performances on multi-step-ahead estimation

As a case, the multi-step-ahead estimation is developed with input combination [$Q_{u(t\text{-}1)}$, $Q_{u(t)}$, $Q_{d(t)}$, $Q_{d(t\text{-}1)}$, $SSL_{u(t\text{-}3)}$] to compute the downstream sediment load at time t. It could be regarded as a three-day-ahead estimation since the sediment input is three days ahead of the output variable. After the optimal model structure has been identified, a thorough comparison of the LUBE PIs constructed by each model is presented in the following.

As indicated in Table 6.5, the reliability of MFNN, DPFNN and HFNN models could not be ensured with 99% confidence level. They are inadequate to generate PIs covering more than 99% observations for the present three-day-ahead estimation. The HDFNN model shows its superiority particularly with this high confidence level, since it could still generate convincing and narrow intervals. When compared with the HFNN, its PIs cover more than 0.28% observations with 2.21% smaller width. The improvement of HDFNN model is not as marked as the one-day-ahead case for the 99% case. The conclusion does not agree with the results of point predictions in Chapter 4, which suggests that the superiority degree of the HDFNN does not significantly mitigate with the increase of lead time. The high confidence level is prone to increase great difficulties for the estimation models in PIs constructions. As far as the 90% and 95% CL cases are concerned, the values of PINAW obtained by the HDFNN model are respectively 1.05% and 1.54%, which is almost the same with the one-day-ahead estimation. However, the current cases produce PIs with smaller coverage probability. Also in light of lower PICP and higher PINAW values obtained by the other three models, it could be concluded that the qualities of PIs substantially deteriorate with increasing lead time. The HDFNN attains narrower intervals than the MFNN, in which the PINAW values are respectively 27.08% and 18.95% smaller for the 90% and 95% CL. The corresponding reduction values for the one-day-ahead case are 22.22% and

13.07%, and comparisons with the above case imply that the HDFNN model could still better present its advantage with larger lead time. This eventually accords with the findings for the point predictions. In other words, the improvements of HDFNN model over the others are greater than the one-day-ahead case with 90% and 95% CL. The main reason might be that the reservoirs assumed in the HDFNN model could better present their storage capability with larger lead time. Furthermore, the HDFNN model is adapted to satisfy the validity condition for the 99% CL for the three-day-ahead case.

The results of partitioning analysis with 95% confidence level are summarized in Table 6.6. The PIs in low and medium loads enclose all corresponding observations while those in high loads are not reliable at 95% CL, which is the same with the results in one-day-ahead case. The current cases generate PIs with even fewer observations enclosed by the HFNN and HDFNN models in terms of high loads. Despite that, the HDFNN model unsurprisingly produces the highest quality of PIs consistently in the three loads. The HDFNN model is better than the HFNN at generating narrower intervals in medium loads for the present case however, which conflicts with the results in one-day-ahead case. This provides another evidence for the excellent performances of HDFNN model in PIs construction with a lead time of three days. That is, although the HFNN model could simulate effective low and medium loads benefitting from the fuzzy pattern-recognition, the ability of HDFNN in modeling the sediment loads with large lead time is preferably presented.

The LUBE generated PIs by various models for the three-day-ahead estimations are demonstrated in Figure 6.6. As can be seen, the four models present inabilities to generate lower bounds consistent with the complexly varied observations, particularly for those in the middle of flow series. As a similar tendency of the comparisons in the

previous case, the DPFNN model produces slightly better upper bounds of PIs than the MFNN. It is observed in Figure 6.6(b) that the superiority of HFNN model over MFNN is obvious in view of the narrower PIs. Moreover, the HFNN is able to produce fluctuating values of upper bounds varying with the observations, owing to the fuzzy patterns. By comparing the results between Figure 6.6(a) and (c), it is also found that PIs attained by the HFNN and HDFNN models are significantly narrower than that by the MFNN and DPFNN. In this context, the HDFNN is better than the HFNN in the generation of upper bounds. It seems that the HDFNN model could not generate PIs with effective lower bounds with a lead time of three days.

Table 6.5 Testing performances by various models for the three-day-ahead case on the Muddy Creek

	models	PICP	CWC	PINRW	PINAW
90% confidence level	MFNN	0.9475	0.0161	0.0161	0.0144
	DPFNN	0.9503	0.0169	0.0169	0.0134
	HFNN	0.9475	0.0159	0.0159	0.0120
	HDFNN	0.9475	0.0130	0.0130	0.0105
95% confidence level	MFNN	0.9668	0.0231	0.0231	0.0190
	DPFNN	0.9668	0.0222	0.0222	0.0180
	HFNN	0.9641	0.0221	0.0221	0.0172
	HDFNN	0.9668	0.0193	0.0193	0.0154
99% confidence level	MFNN	0.9834	0.0491	0.1321	0.0431
	DPFNN	0.9862	0.0496	0.1169	0.0427
	HFNN	0.9889	0.0431	0.0900	0.0362
	HDFNN	0.9917	0.0443	0.0443	0.0354

Table 6.6 Statistics of different loads by various models for the three-day-ahead case on the Muddy Creek at 95% confidence level

Model	Low load			Medium load			High load		
	PICP	PINRW	PINAW	PICP	PINRW	PINAW	PICP	PINRW	PINAW
MFNN	1	0.0103	0.0103	1	0.0234	0.0216	0.8310	0.0402	0.0373
DPFNN	1	0.0100	0.0100	1	0.0211	0.0189	0.8310	0.0395	0.0373
HFNN	1	0.0079	0.0072	1	0.0242	0.0228	0.8169	0.0380	0.0341
HDFNN	1	0.0076	0.0071	1	0.0187	0.0180	0.8310	0.0348	0.0327

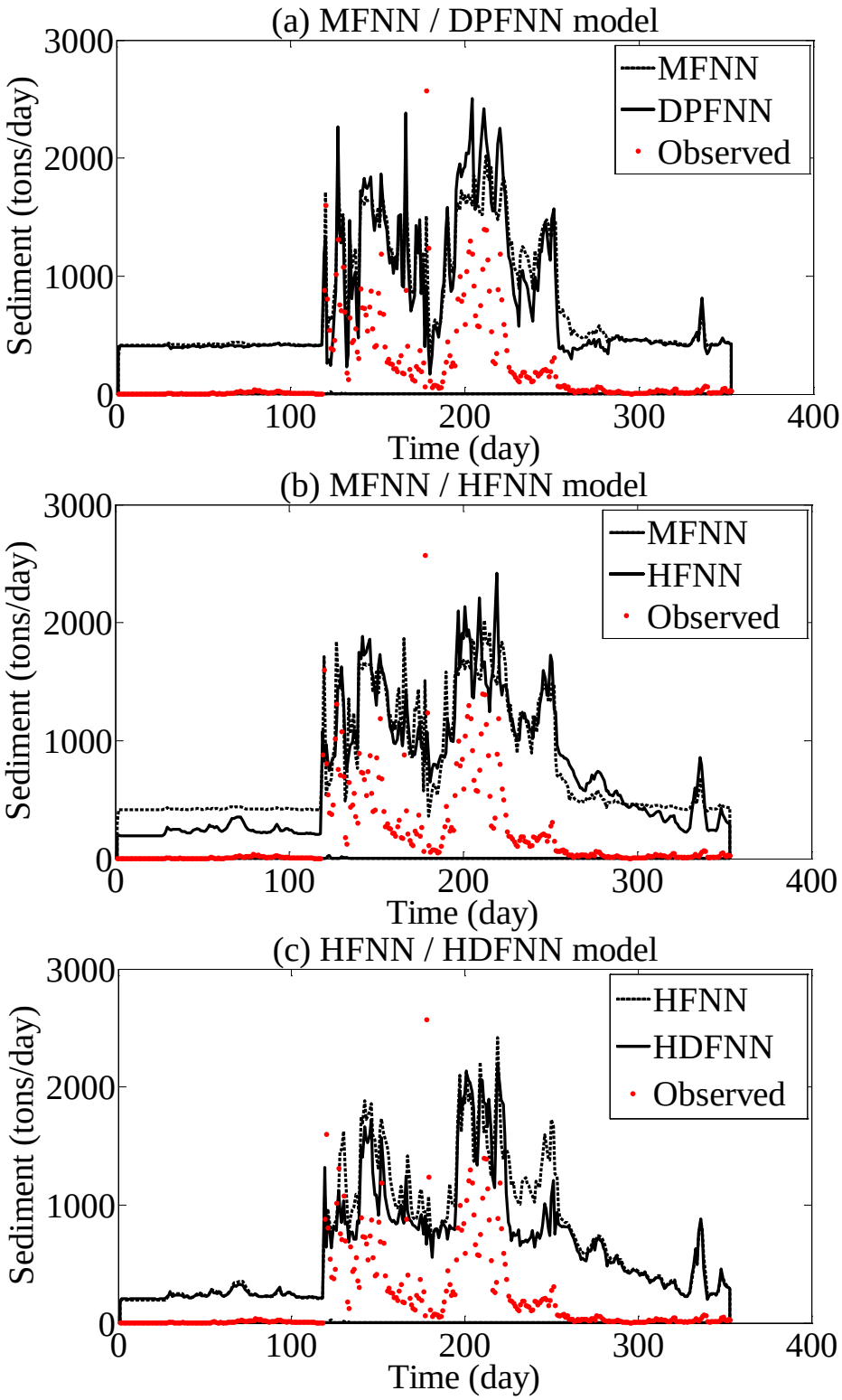

Figure 6.6 LUBE generated PIs at 95% confidence level for three-day-ahead case on the Muddy Creek by (a) MFNN/DPFNN (b) MFNN/ HFNN and (c) HFNN/HDFNN models

6.5 Summary

The uncertainty analysis on hydrological models for river flow forecasting and sediment load estimation are undertaken respectively in this chapter, as a complementary study to the research in Chapters 3 and 4. Instead of examining the model effectiveness, the main objective of this study is to assess the reliability of models by interval predictions and finally evaluate their applicability in practical use. The LUBE is a straight-forward method that can construct neural network based models with two output neurons and directly approximate the lower and upper bounds of PIs. An index, namely Coverage Width-based Criterion (CWC), as a combination of coverage probability and interval width, is employed in the LUBE model in order to search high quality PIs.

Applications on the river flow forecasting models reveal the reliability of HNN model from the perspective of one-day-ahead and 16-hour-ahead cases with 90% and 95% confidence level. In regard to the 99% CL case, the HNN model generates valid and narrow PIs with a lead time of one day, although the improvements over the other three models are not as significant. The large lead time in the 16-hour-ahead case increases the uncertainties in the inputs, and thus leads to failure to ensure the required coverage probability with confidence level 99%.

As for the sediment load estimation, the superiority degree of the HDFNN over its counterparts does not significantly mitigate with the increase of lead time for the 90% and 95% CL cases. The more compact structure of HDFNN model renders narrower PIs entailing less uncertainty. Further, it could produce better lower bounds of PIs owing to the integration of the continuity equation and the employment of recession coefficients. The fuzzy patterns in the HDFNN model could explain the generated PIs with better

upper bounds. Partitioning analysis reveals that the HDFNN consistently constructs narrow PIs with 100% enclosed observations in low and medium loads, although with invalid PIs in high loads with 95% CL. With confidence level 99%, it is more difficult to optimize the parameters from the input to the hidden layer to adapt both the lower and upper bounds of the PIs, particularly when more uncertainties are imported with increasing lead time. Only the HDFNN could ensure the required coverage probability and generate the narrowest PIs among the four models for the three-day-ahead case.

The influences of hydrological models on the quality of the PIs are found to be significant in this study using DE algorithm to minimize the objective function. Further research should be focused on examining the effectiveness of other optimization algorithms in addressing the highly nonlinear, discontinuous and non-differentiable problem in LUBE method. In addition, the identification of optimal input variables for the interval predictions should be undertaken using appropriate input selection techniques.

7 Comparative Study on Population-based Optimization Algorithms

The effectiveness and reliability of population-based optimization algorithms in neural network models are another concern in this thesis. The optimization confronted with the HNN and HDFNN models is characterized as a highly-nonlinear and non-differentiable problem, which entail the use of population-based optimization algorithms. The present chapter primarily concentrates on the following contents:

(1) To compare the optimization ability, convergence speed and reliability of three population-based optimization algorithms (i.e., DE, ABC and ACO);

(2) To determine the most adaptive population-based optimization algorithm for the HNN and HDFNN models.

7.1 Introduction

Traditional optimization algorithms namely the gradient descent method and Levenberg-Marquardt (L-M) method are not applicable for non-differentiable problems. Besides, local optima are more likely to happen when the models to be optimized are characterized with multi-dimension, high nonlinearity and varying with time. Recently, population-based optimization algorithms have attracted the interest of many researchers (Blum and Roli, 2003; Chiong et al., 2012). They are characterized by a population consisting of possible solutions to the problem, which are modified by applying different types of operators and thus moving towards a near-optimal solution area. These algorithms are very powerful in finding a global optimum since they

simultaneously search in many directions by using a population of possible solutions. Generally, there are two categories of population-based optimization algorithms: evolutionary algorithms and swarm intelligence algorithms (Blum et al., 2012). Typical methods of evolutionary algorithms are the genetic algorithm (GA) and differential evolution (DE), which attempt to simulate natural evolution. The DE algorithm was proposed by Storn and Price (1995) and has been applied extensively in hydrological modeling (e.g., Babu and Angira, 2003; Liu and Pender, 2013; Vasan and Simonovic, 2010). It shows better performances than the GA in terms of convergence characteristics and computational efficiency (Li et al., 2013a; Song et al., 2014; Wang et al., 2008b). In view of its ability to handle optimization problems that are non-differentiable, nonlinear, non-continuous and varying with time (Rocca et al., 2011), it is adopted in this chapter for the comparative study.

The second category, swarm intelligence-based algorithms are inspired by the collective behavior of animal societies, including particle swarm optimization (PSO), artificial bee colony (ABC) and ant colony optimization (ACO). This chapter focuses on ABC and ACO as stochastic global optimization algorithms. The ABC algorithm was introduced and popularized by Karaboga (2005) to solve numerical optimization problems. It has predictive capability comparable to the GA, PSO and DE algorithms on numerical test functions (Karaboga and Akay, 2009). Hybrid models that combine ABC algorithms with ANNs have been developed recently (Karaboga et al., 2007; Kisi et al., 2012b). Another swarm intelligence-based optimization method ACO was derived from the food searching behavior of ants (Dorigo et al., 1996). Similar to the ABC algorithm, it is a meta-heuristic technique available to solve non-linear optimization problems with high dimensionality and inequality constraints. Coupling ACO algorithm with feed-forward neural network training has proven to be successful (Li and Chung, 2005; Shelokar et al.,

2007; Socha and Blum, 2007). The potential to apply ACO to the field of river flow forecasting and sediment load estimation is clear, e.g., see its applications in water resource problems (Jalali et al., 2006; Kumar and Reddy, 2006; Maier et al., 2003).

Optimization of neural networks has always been an open research. It is imperative to solve the disadvantages of traditional learning algorithms, such as poor generalization, slow convergence speed and easily plunging into local optima. The main objective of this chapter is therefore, to incorporate population-based optimization algorithms (namely DE, ABC and ACO) into the HNN and HDFNN models, and compare their optimization ability, stability and reliability, and thereby, determine the most adaptive optimization algorithm for the river flow forecasting and sediment load estimation. The rest of this chapter is structured in the following manner. In the section 7.2, a brief review of the three population-based optimization algorithms is firstly provided. Section 7.3 then applies the algorithms into the river flow forecasting model, with single- and multi-step-ahead cases respectively. The applications and performances of the optimization algorithms on the HDFNN model are subsequently presented. Finally, the computational results and comparisons of the optimization algorithms are summarized in the section 7.5.

7.2 Population-based optimization algorithms

7.2.1 Differential evolution algorithm

The differential evolution (DE) algorithm conducts mutation, crossover and selection operations based on the differences of randomly sampled pairs of solutions in the population. Mutation operation acts as a search mechanism while the crossover operator recombines the parent vector with the mutated one. Afterwards, all solutions have an

equal chance of being selected as a parent. The strategies are described as follows.

Firstly, define a population of D dimensional parameter vectors, with NP population size. Each individual (target vector) for generation G is then represented as $X_{i,G}$ (i =1, 2, …, NP) and $X_{i,G} = \{x_{i1,G}, x_{i2,G}, ..., x_{iD,G}\}$ constitutes all parameters to be optimized. Donor vector in the next generation $V_{i,G+1}$ is generated from three randomly selected vectors $X_{r1,G}$, $X_{r2,G}$ and $X_{r3,G}$ as follows

$$V_{i,G+1} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) \tag{7.1}$$

in which F acts as a mutation factor which is a random number uniformly distributed within the range [0, 2]. The randomly selected indexes $r1$, $r2$, $r3 \in \{1, 2, \ldots, NP\}$ must be different from one another and from the running index i as well.

The crossover operation defines a trial vector $U_{i,G+1} = \{u_{i1,G+1}, u_{i2,G+1}, ..., u_{iD,G+1}\}$ as follows

$$u_{ij,G+1} = \begin{cases} v_{ij,G+1} & \text{if } rand_{ij} \le CR \text{ or } j = I_{rand} \\ x_{ij,G} & \text{if } rand_{ij} > CR \text{ and } j \ne I_{rand} \end{cases} \tag{7.2}$$

where j=1, 2, …, D; CR is a crossover constant in the range of [0, 1]; $rand_{ij}$ is a random number within 0 and 1 ; and I_{rand} is a random index from [1, 2, …, D], ensuring that at least one element in the trial vector is obtained from $V_{i,G}$. The change of the diversity of the population is controlled by the CR value.

The selection operation chooses the vector in the next generation by the following equation for a minimizing problem

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } f(U_{i,G+1}) \le f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \tag{7.3}$$

in which the objective function associated with $X_{i,G}$ is denoted as $f(X_{i,G})$. In other words, when comparing the target vector with the trial vector, a lower function yielding value is admitted to the succeeding generation. In general, the DE algorithm is a comparatively simple algorithm because only three control parameters (i.e., *NP*, *F* and *CR*) are required to be prescribed.

7.2.2 Artificial bee colony

The ABC algorithm draws its inspiration from the foraging behavior of honey bee swarms. A possible solution to the optimization problem is viewed as a food source for the artificial bees. In the foraging process, bees interact with each other aiming to maximize the nectar amount of the food source. The distribution and duties of grouped employed, onlooker and scout bees have been described in details (Karaboga and Akay, 2009).

In this scheme, a population of food source position $\{X_1, X_{2, \ldots,} X_{NP}\}$ is initialized randomly, where *NP* denotes the number of population and refers to the number of food sources as well. Each food source $X_i = \{x_{i1}, x_{i2}, ..., x_{iD}\}$ is a *D*-dimensional vector, containing *D* variables for the optimal problem. During the employed bees' phase, a neighbor solution v_{ij} is generated from an original one x_{ij} by the following equation

$$v_{ij} = x_{ij} + rand(-1,1) \times (x_{ij} - x_{kj}) \tag{7.4}$$

where the subscript $j \in [1, D]$ are randomly chosen indexes and $k \in [1, NP]$ is a random neighbor index which should be different from *i*. After evaluating the new neighbor and original solution by the fitness value of the optimal problem, the better solution will be kept in the population. Employed bees will then transmit the information concerned with the source to the onlooker bees. The onlooker bees employ a roulette-wheel like

selection depending on a probability value as follows

$$p_i = \frac{f(X_i)}{\sum_{k=1}^{NP} f(X_k)} \tag{7.5}$$

where $f(X_i)$ is the fitness value of the solution X_i. The probability value p_i is proportional to the fitness value for a selected solution. If p_i is larger than a random number drawn for each solution within [0, 1], the onlooker bees will make a local search as in Eq. (7.4) to find a neighbor solution. During the employed and onlooker bees' phases, when the nectar of a food source being exploited is exhausted, the source will be abandoned. That is, if the original solution is kept and not replaced by a new neighbor one for exceeding a prescribed number of cycles, namely *Limit*, it is postulated to be an exhausted one and has to be replaced by a new random solution. The scout bees determine a new food source by the following expression:

$$x_{ij} = x_j^{min} + rand(0,1) \times (x_j^{max} - x_j^{min}) \tag{7.6}$$

in which, x_j^{min} and x_j^{max} are the minimum and maximum value of the corresponding solution, respectively. Similarly, the new solution will be evaluated and compared with the existing one. The solution with a "rich source" (minimum fitness value) will be selected as the optimal one. There are two control parameters in an ABC algorithm: *NP* and *Limit*.

7.2.3 Ant colony optimization

The ACO algorithm simulates ants' behavior in which their fundamental objectives are to find the shortest path between food source and their nest. There is a hypothetical chemical substance named pheromone laid by other ants in their trails, which works as a communication mechanism. The ants will choose paths independently according to the

pheromone intensity. For a specific solution component, it is more likely to be selected if its pheromone value is higher. The pheromone value will be reinforced if the corresponding solution component belongs to the best solution. The following discussion outlines the basic concept of the ACO algorithm.

Given that there are D parameters to be optimized in the search space. Each parameter's definition space is split into a set of discrete points. As shown in Table 7.1, for parameter x_j (j=1, 2, …, D) within a range of $[a_{j1}, a_{jN}]$, there are totally N discrete points when the feasible range is uniformly divided into N-1 shares. Each point corresponds to a candidate value of the parameter, namely solution component a_{jk} (k=1, 2, …, N). Suppose an ant i (i=1, 2, …, NP) is on its path to search D parameters, it can only choose a value for each parameter among the candidate points and record the corresponding tag. In the meantime, the pheromone intensity is needed for each candidate point, represented by τ_{jk} for tag k. When an ant reaches the parameter x_j, each solution component has the following probability to be selected

$$p_{jk} = \frac{\tau_{jk}}{\sum_{1 \le m \le N} \tau_{jm}} \tag{7.7}$$

With probability value larger than a random value within the range [0, 1], the corresponding point a_{jk} will be selected for this parameter. After the ant finishes its tour and all parameters are selected as $X_i=\{x_1, x_2, \ldots, x_D\}$, it will return to its nest and update the pheromone intensity according to the following equations:

$$\tau_{jk}(i+1) = \rho \tau_{jk}(i) + \Delta \tau_{jk} \tag{7.8}$$

$$\Delta \tau_{jk} = \begin{cases} Q / f(X_i) & \text{if } a_{jk} \text{ is selected as } x_j \text{ and belongs to } X_i \\ 0 & \text{else} \end{cases} \tag{7.9}$$

The first term in the right-hand side of Eq. (7.8) reflects an evaporation process in which $\rho \in [0, 1]$ is the coefficient of pheromone duration. The second term indicates the

reinforcement in which $\Delta\tau_{jk}$ is the amount of pheromone retained in the solution component being a part of X_i. The reinforcement value $\Delta\tau_{jk}=0$ if component a_{jk} is not within the best solution. In particular, the pheromone constant Q in Eq. (7.9) is a user-defined parameter which is the same for all ants, and $f(X_i)$ is the fitness value associated with X_i. The parameters of the ACO algorithm to be prescribed are respectively NP, N, ρ and Q.

Table 7.1 Pheromone table for each parameter in ACO algorithm

x_j (j=1, 2,…, D)					
Tag	1	2	…	N-1	N
Solution component	a_{j1}	a_{j2}	…	$a_{j(N-1)}$	a_{jN}
Pheromone intensity	τ_{j1}	τ_{j2}	…	$\tau_{j(N-1)}$	τ_{jN}

This chapter employs the population-based algorithm to deal with the non-differentiable and combinatorial problems of the HNN and HDFNN models. Given that the number of parameters to be optimized is D, this set of parameters is represented by a vector X_i. As outlined in Figure 7.1, the steps for optimizing parameters, when taking the DE algorithm as an example, are as follows: (1) randomly generate a population $\{X_1, X_{2,\ldots,} X_{NP}\}$ within their interval $[x_j^{min}, x_j^{max}]$; (2) define the objective function of the model and set the running generation G=1; (3) train the model with current updated parameters and obtain the corresponding fitness value of the objective function; (4) apply the mutation and crossover processes to construct donor and trial vectors, and finally select one vector with a minimum fitness value from the trial and target vectors; (5) reset G=G+1; (6) if G>MI (maximum iteration), the stopping criterion is satisfied and the optimal set of parameters is outputted; or go to step (3) and (4) for the next generation. In particular, for step (4), the configuration of ABC and ACO algorithms is demonstrated in Figure 7.1 as well. The ABC algorithm undergoes the employed, onlooker and scout bees' phases orderly. As for the case of the ACO algorithm, firstly, set i=1 and initialize equal

pheromone intensity; release ant X_i, and let it select a candidate value for all D parameters; update the pheromone intensity and reset i=i+1; if i<=NP, release the succeeding ant and begin its tour, otherwise all ants have finished the search. All these three techniques show an easy realization in the process of optimizing parameters for the HNN and HDFNN models.

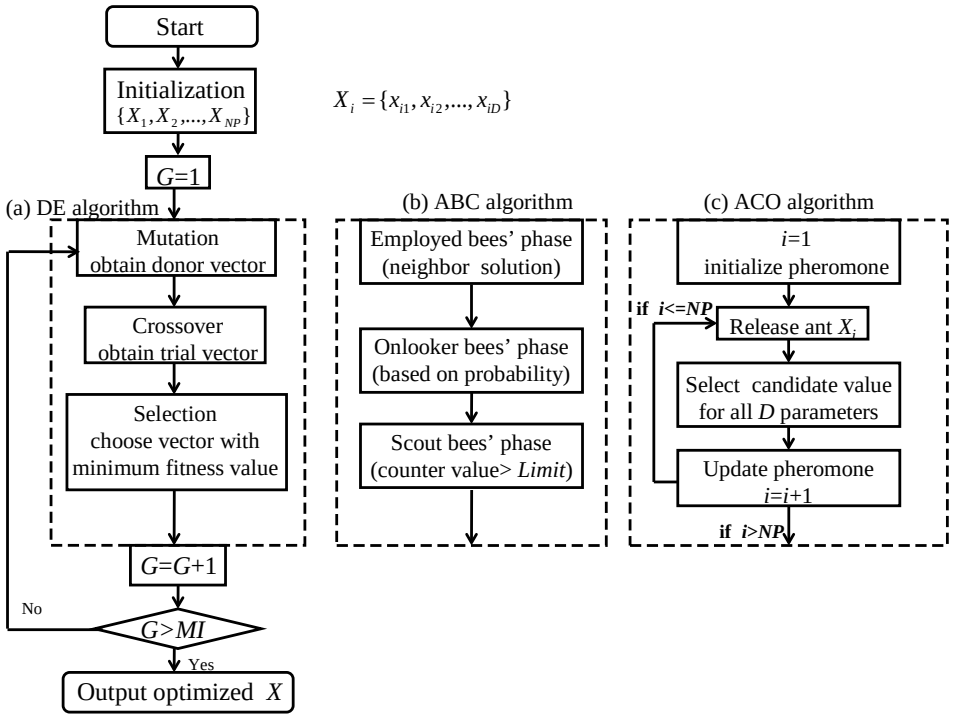

Figure 7.1 Framework for optimizing parameters by (a) DE (b) ABC (c) ACO algorithm

7.3 Applications on river flow forecasting

In order to compare the performances of the population-based optimization algorithms on the HNN model, the Yellow River in the Altamaha river basin is employed as a case study site. This chapter adopts the same daily and hourly data to compute the river flow at the downstream station 02208000. When the optimization ability, convergence speed and stability of each algorithm might be relevant with forecast lead time, both single- and multi-step-ahead cases are performed in the following.

7.3.1 Performances on single-step-ahead forecasting

As been shown in Chapter 5, the DE algorithm yields the best forecasting performances when daily areal precipitation and all three upstream flows are employed as input variables. These results are compared with that optimized by the ABC and ACO algorithms using the same input. For completeness' sake, the PSO algorithm is included as a benchmark case for the comparison of forecasting performances, a detailed description of which is given by Clerc and Kennedy (2002).

Firstly, the training and testing performances by various algorithms in terms of RMSE, NSEC and ACC are summarized in Table 7.2. The DE algorithm outperforms its counterparts in both training and testing period with respect to RMSE and NSEC values. The high values of NSEC suggest that the HNN model is fully trained to provide sufficiently accurate forecasting by the DE algorithm, although obtaining a lower ACC value than the PSO in the testing stage. The advantage of DE over ABC algorithm is relatively evident, as there is a 9.76% reduction in RMSE and 4.92% increase in NSEC value during the testing period. When compared with the PSO algorithm, ABC does not show superiority in this study case. Comparison of performances between DE and ACO implies that the ACO algorithm can be recommended as an effective method as well. It has comparable performances with the PSO algorithm, particularly for the RMSE and NSEC values in the training period. From the perspective of generalization and forecasting abilities, the HNN model superbly captures the input-output mapping when optimized with DE and ACO algorithms. To obtain the above results, parameters used are: MI =500, NP=550, F=0.4 and CR=1.0 in the DE algorithm; MI =2000, NP=440 and $Limit$=214 in the ABC algorithm; MI =500, NP=780, N=45, ρ=0.3 and Q=1.0 in the

ACO algorithm.

The scatter plots of observed and computed discharges by these four algorithms are presented in Figure 7.2. The HNN model exhibits good matches between the observed and computed data series in the testing period. The plots of intensively distributed dots along the ideal line within 0-50 m^3/s suggest that the low river flows are mostly well forecasted. This is because the frequent occurrences of low values allow a better generalization of the trained model. The performance of the ABC algorithm is not as good when compared with the DE, ACO and PSO with evidence of five apparent dots over-forecasted with a range of 113-226 m^3/s. The result is consistent with the high RMSE value obtained by the ABC algorithm in Table 7.2. The improvement of performances for the DE over ACO algorithm is not evident in the scatter plots.

The capability of capturing extreme values is critical to evaluate the performance of the HNN model. Figure 7.3 presents the time series of observed and computed discharges by the DE algorithm and marks five observed extreme points. The quantitative values are provided in Table 7.3, along with the forecasted results by the four algorithms. Whilst the peak value (extreme point 2) is over-forecasted by all algorithms, the DE algorithm produces a relatively closer result. It performs well at extreme point 3 although the forecasted one is slightly under-estimated. The ACO algorithm gives the most approximate values for point 4 and 5. The relative mean errors between the computed and observed extreme values by all four algorithms do not vary significantly. Accordingly, the four algorithms perform equally well in capturing extreme values. They have operators providing variable step size and diversity, thus can perform the perturbation of the proposed HNN model.

Table 7.2 Training and testing performances by various algorithms in terms of RMSE, NSCE and ACC for the one-day-ahead case in the Altamaha river basin

Algorithm	Training			Testing		
	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC
DE	**2.5534**	**0.9812**	0.7627	**8.1500**	**0.8294**	0.7937
ABC	3.0827	0.9727	0.7279	9.0319	0.7905	0.7905
ACO	2.8174	0.9772	**0.7658**	8.8573	0.7985	0.7853
PSO	2.8306	0.9769	0.7139	8.1502	**0.8294**	**0.8150**

Table 7.3 Observed and computed extreme values by various algorithms for the one-day-ahead case in the Altamaha river basin

Algorithm	Extreme point 1 (m^3/s)	Extreme point 2 (m^3/s)	Extreme point 3 (m^3/s)	Extreme point 4 (m^3/s)	Extreme point 5 (m^3/s)	Relative mean error
Observed	89.4811	156.8751	69.6593	49.8376	124.0276	
DE	66.4992	**219.9800**	**68.8210**	64.1180	117.3200	**0.2024**
ABC	**75.2180**	223.3100	67.5240	69.5910	132.1700	0.2151
ACO	54.3910	224.1600	71.0100	**61.9400**	**125.1600**	0.2184
PSO	71.8230	223.7800	62.4530	64.5090	115.9200	0.2174

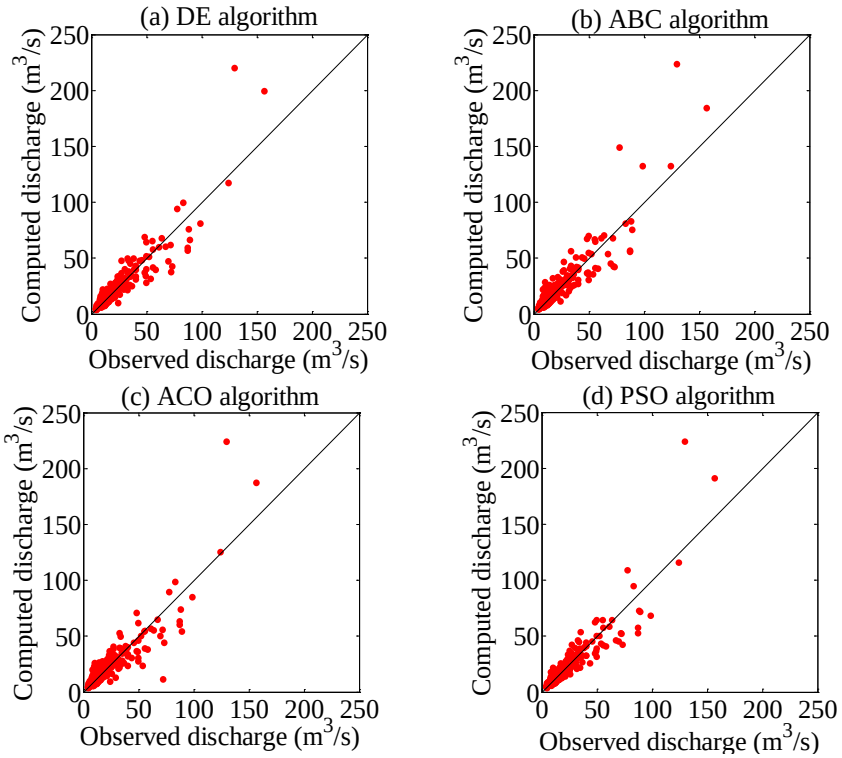

Figure 7.2 Observed and computed discharges by (a) DE (b) ABC (c) ACO and (d) PSO algorithm for the one-day-ahead case in the Altamaha river basin

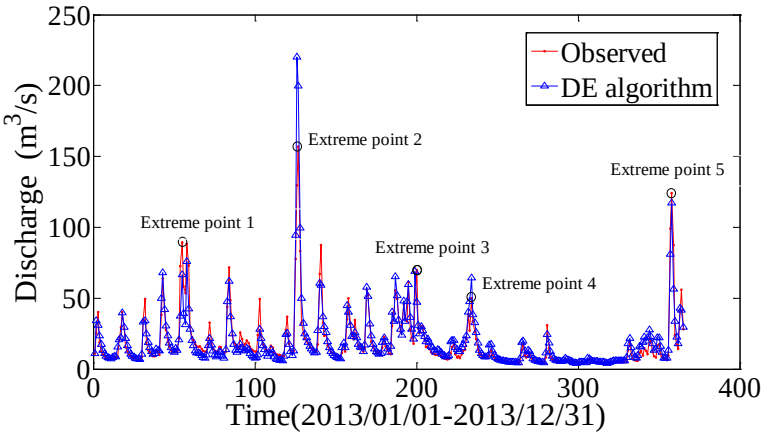

Figure 7.3 Observed and computed discharges by DE algorithm and marked extreme points for observed discharges for the one-day-ahead case in the Altamaha river basin

Details regarding the mean, variance and mean of 95% confidence interval by the four algorithms are provided in Table 7.4. This statistical analysis is necessary to check the significance of the differences. The mean value attained by the PSO algorithm is closer to the observed one when compared with the others. These four algorithms obtain higher variance values than the observations, which indicate that the computed results are more widely distributed. The most approximated value for the observations is obtained by the PSO algorithm for the mean with 95% confidence interval, yet slight discrepancy is observed in comparison with other algorithms. Thus the forecasting performances by these four optimization algorithms are comparable.

The Diebold-Mariano (DM) test is employed to assess the statistical significance of the comparison of the results. It is used to compare the accuracy of two forecasts, by computing the DM statistics on the base of the loss differential defined as the difference of the squared forecast errors (Diebold and Mariano, 1995; Harvey et al., 1997; Rech, 2002). Under the null hypothesis of equal forecasting accuracy, the asymptotical distribution of DM is standard normal. Suppose that the significance level of the test is 0.05, the null hypothesis of no difference will be rejected if the computed DM statistic

falls outside the range of -1.96 to 1.96. Table 7.5 provides the DM statistics between each pair of algorithms. Observations reveal that all the DM statistics are within the range of [-1.96, 1.96], which give all the algorithms equal chance of forecast. This supports the reliability of the comparative study of optimization algorithms. The DM statistics between DE and the other three algorithms are -1.0389, -1.4719 and -0.0005 respectively, and these negative values reveal a preference with regard to the DE algorithm. Inversely, the positive DM statistics between ABC and the other three imply the inferiority of the former algorithm. The PSO and DE algorithms show almost equal accuracy, as indicated by the low DM statistic (i.e., 0.0005). These results accord with the above discussions on forecasting performances.

Table 7.4 Statistical analysis by various algorithms for the one-day-ahead case in the Altamaha river basin

Algorithm	Mean (m^3/s)	Variance (m^6/s^2)	Mean of 95% confidence interval (m^3/s)
Observed	18.6845	389.4509	15.1565
DE	19.2846	453.6750	16.3556
ABC	19.6943	512.7502	16.1543
ACO	18.8421	433.9878	16.0703
PSO	18.5927	441.1913	15.8193

Table 7.5 DM test for the forecasting accuracy between each pair of algorithms for the one-day-ahead case in the Altamaha river basin

Algorithms	DE	ABC	ACO	PSO
DE		-1.0389	-1.4719	-0.0005
ABC	1.0389		0.1905	1.2904
ACO	1.4719	-0.1905		1.1300
PSO	0.0005	-1.2904	-1.1300	

In addition, the convergence speed of population-based optimization algorithms is investigated, in which the training error as a function of running iteration is illustrated in Figure 7.4. The training error decreases rapidly when the number of iterations is less than 200 and remains approximately constant when the running iteration approaches 500 times for the DE and ACO algorithms. That is, these two algorithms can achieve

convergence with 500 time runs. The PSO algorithm has faster speed since the training error decreases dramatically and becomes stable before running 200 times. In contrast, the ABC algorithm reaches the minimum training error until the running iteration approaches 2,000. However, the minimum training errors produced by the four algorithms are sufficiently small, indicating satisfactory precision of the generalization. These four algorithms have a powerful global search ability to prevent the optimization from the premature convergence. As shown in Table 7.6, the running times for each algorithm are respectively 3.12, 29.4, 4.43 and 1.64 hours. The ABC algorithm displays slow convergence speed, and thereby, is time consuming. This is mainly attributed to the localization of the method itself, in which two phases with equally large population of bees are assigned in the searching process. The values of special parameters, namely recession coefficient and initial storage, are also presented in the table for comparison. The ABC algorithm obtains quite different values while the other three tend to attain reliable values. Thereby, the DE and ACO algorithms have the feature of rapid convergence on global optima and high precision in searching parameters of the HNN model. They are consistently better than the ABC algorithm in time and derivation efficiency.

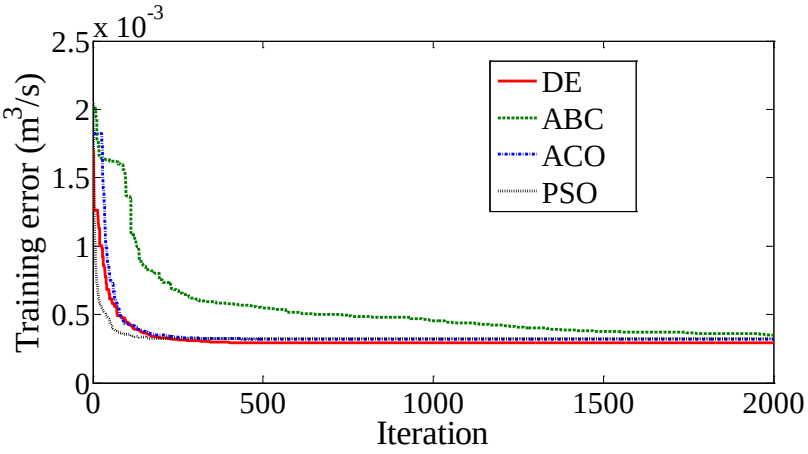

Figure 7.4 Training errors (RMSE) by various algorithms versus running iteration for the one-day-ahead case in the Altamaha river basin

Table 7.6 Running time and values of special parameters by various algorithms for the one-day-ahead case in the Altamaha river basin

Algorithm	Maximum iteration (*MI*)	Running time (hour)	Parameters	
			recession coefficient λ	initial storage $S_{(T=0)}$
DE	500	3.12	0.3609	0.1327
ABC	2000	29.4	0.3032	0.0532
ACO	500	4.43	0.3864	0.1365
PSO	500	1.64	0.3661	0.1309

A comprehensive comparative study on the stability of DE, ABC and ACO algorithms is further carried out. The parameter of population size *NP* poses a great impact on the performance of population-based optimization algorithms. A large *NP* can endow the algorithms with powerful ability of exploring more possible solutions, albeit computationally intensive. Figure 7.5 presents the influence of *NP* on the RMSE value by various algorithms. The DE algorithm has an adaptive relation between *NP* and RMSE, where the RMSE value firstly decreases and remains stable with increasing *NP*. The choice of *NP* assures the stability of the DE algorithm. This is most likely due to donor and trial vectors, which provide diversity of population and thus weaken the dependency of the performance on population size. The ABC and ACO algorithms could not reveal such consistency, since the *NP* significantly affects the RMSE value. Nevertheless, the variation of RMSE with *NP* is within a reasonable range.

Mutation factor *F* and crossover constant *CR* serve as the other two control parameters of the DE algorithm, and their influences on the RMSE value are demonstrated in Figure 7.6. The results of Figure 7.6(a) are obtained with fixed *CR*=1.0 while those for Figure 7.6(b) are obtained with fixed *F*=0.4. The best RMSE value is attained when *F* and *CR* are recognized as 0.4 and 1.0, respectively. However, the plots of RMSE versus *F* and *CR* are quite different. As *F* increases from 0 to 2, RMSE first drops to a minimum value and then begins to increase. This implies that RMSE is sensitive to *F*

for the DE algorithm. As a basic requirement, the chosen *F* should be able to provide a diversity of mutated vectors. The RMSE decreases sharply from 0 to 0.1, and fluctuates within a narrow range when *CR* increases. The DE algorithm presents a certain degree of stability in terms of *CR* within the interval between [0.1, 1.0].

Limit is an inclusive control parameter for the ABC algorithm, which governs the number of scout bees' exploration. As illustrated in Figure 7.7, the value of *Limit* is recommended to be smaller than the number of food source *NP* since a too high value of *Limit* will lead to an ineffective scout bees' phase. A too small *Limit* value which indicates that employed and onlooker bees do not work is not appreciated as well. In an ABC searching process, both exploration by scout bees and exploitation by employed and onlooker bees must be efficiently carried out (Karaboga, 2005). In this simulation, the ABC algorithm presents stability in terms of *Limit* in the interval of [80, 320]. A slight advantage of the ABC over the other two algorithms is that there is only one control parameter in addition to *NP*.

Figure 7.8 presents the sensitivities of RMSE value to the number of solution components *N*, pheromone duration coefficient ρ and pheromone constant *Q* respectively for the ACO algorithm. The precision of finding the optimal solution is low and the best solution may not be obtained if *N* is small and the amount of available solutions is insufficient. On the contrary, a larger value of *N* does not definitely ensure better searching results even though there are more solution components to be selected. As displayed in Figure 7.8(a) with fixed ρ=0.3 and *Q*=1.0, the RMSE value strongly depends on *N* with irregular fluctuations. Generally, *N* can be enacted to be less than 100 for this case since the RMSE varies within a small range. The fixed parameters given in Figure 7.8(b) are *N*=45 and *Q*=1.0 whilst those in Figure 7.8(c) are *N*=45 and

ρ=0.3. The ACO algorithm exhibits stability regarding ρ and Q, apart from infeasible cases both on the boundary value 0. An appreciable value of ρ enables the ant to effectively forget the fallacious solution component and explore a new one. The value of Q is associated with the fitness value, and a range of [0.1, 1] is suggested to assure the stability of ACO algorithm in this study. However, these values for the control parameters vary with the dimensions of the problem and other specific characteristics. The recommended values to these parameters may not be applicable in other studies.

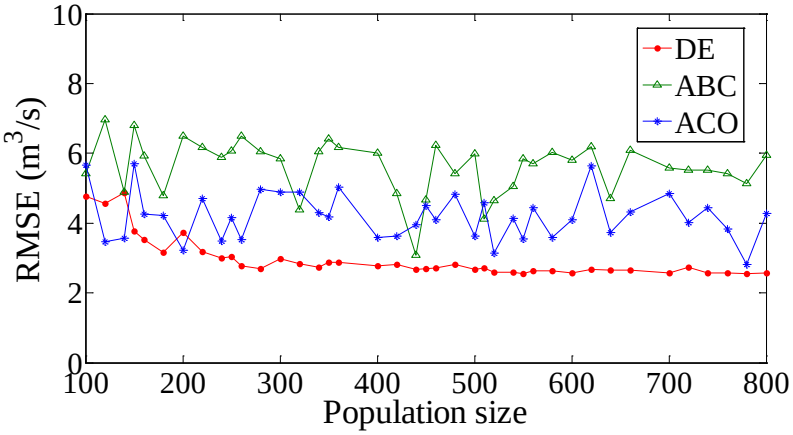

Figure 7.5 The influence of population size on forecasting performances by various algorithms for the one-day-ahead case in the Altamaha river basin

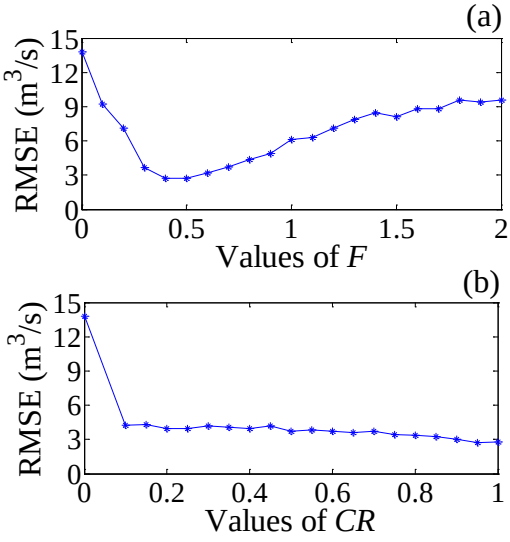

Figure 7.6 The influences of (a) F (b) CR on forecasting performances by DE algorithm for the one-day-ahead case in the Altamaha river basin

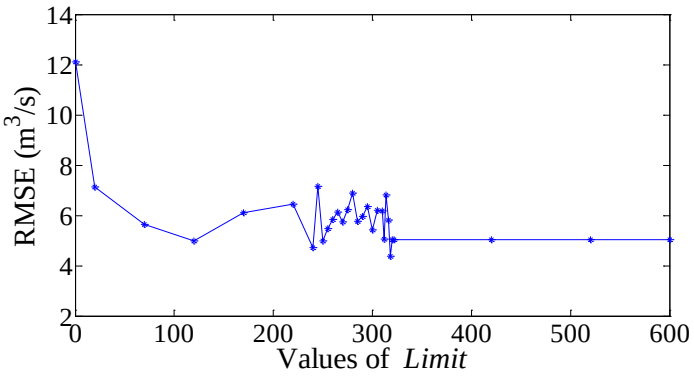

Figure 7.7 The influence of *Limit* on forecasting performances by ABC algorithm for the one-day-ahead case in the Altamaha river basin

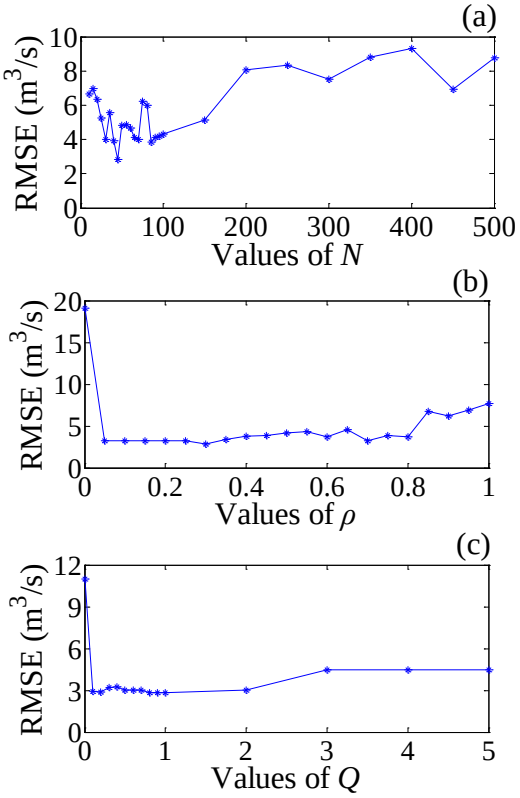

Figure 7.8 The influences of (a) N (b) ρ (c) Q on forecasting performances by ACO algorithm for the one-day-ahead case in the Altamaha river basin

In conclusion, the DE algorithm presents the best performance on the generalization and forecasting for the HNN model, which is quite comparable to the PSO algorithm. The ACO is a feasible algorithm yielding satisfactory forecasting results as well. The ABC algorithm does not exhibit a comparable efficiency, with slow convergence and time consumption. The ACO and DE algorithms are found to give a consistent parameter optimization of the HNN model in terms of the recession coefficient and initial storage.

Also in view of stable performances regarding their control parameters, the DE and ACO algorithms are well adapted for the optimization problem in the HNN model.

7.3.2 Performances on multi-step-ahead forecasting

As far as the performance of optimization algorithms on multi-step-ahead forecasting is concerned, the 16-hour-ahead forecast in the Altamaha river basin is presented in this section. The DE algorithm has shown its efficiency with a high NSEC value (i.e., 0.8929) from input combination 2-1 in Chapter 5. The training and testing performances by the other algorithms are shown in Table 7.7 based on the same hourly data and input. As can be seen, the DE attains slightly inferior results than the PSO, yet they are perfectly capable algorithms with large forecast lead time. The ABC algorithm yields the best RMSE and NSEC values in the training period, revealing its ability in generalization. The forecasting accuracy by the ABC algorithm is fairly appropriate as well, in which the ACC value is slightly higher than that by the DE algorithm. Thus the ABC algorithm could be employed as an effective optimization technique with the HNN model when multi-step-ahead forecast is performed. The ACO algorithm consistently yields comparable results in this study case, and obtains the best accuracy in particular. The corresponding forecasted results are plotted in Figure 7.9 to provide a visual comparison. The computed dots by the four algorithms are basically distributed along the ideal lines, indicating that they are alternative methods for the hourly forecasting. Some of the dots in the range of 10-30 m^3/s are perfectly matched with the observations. Besides, the ACO algorithm slightly outperforms others with evidence of intensively distributed dots within the range of 30-50 m^3/s. The above results are achieved by the following parameters: *MI* =500, *NP*=600, *F*=0.4 and *CR*=0.9 in the DE algorithm; *MI* =2000, *NP*=460 and *Limit*=226 in the ABC algorithm; *MI* =500, *NP*=800, *N*=500, ρ=0.3

and Q=0.1 in the ACO algorithm.

Table 7.7 Training and testing performances by various algorithms in terms of RMSE, NSCE and ACC for the 16-hour-ahead case in the Altamaha river basin

Algorithm	Training			Testing		
	RMSE (m^3/s)	NSEC	ACC	RMSE (m^3/s)	NSEC	ACC
DE	2.6799	**0.9763**	**0.9327**	2.8348	0.8929	0.8643
ABC	**2.6792**	**0.9763**	0.9320	2.8377	0.8927	0.8667
ACO	2.7990	0.9742	0.9266	2.9636	0.8830	**0.8865**
PSO	2.7082	0.9758	0.9299	**2.8114**	**0.8947**	0.8819

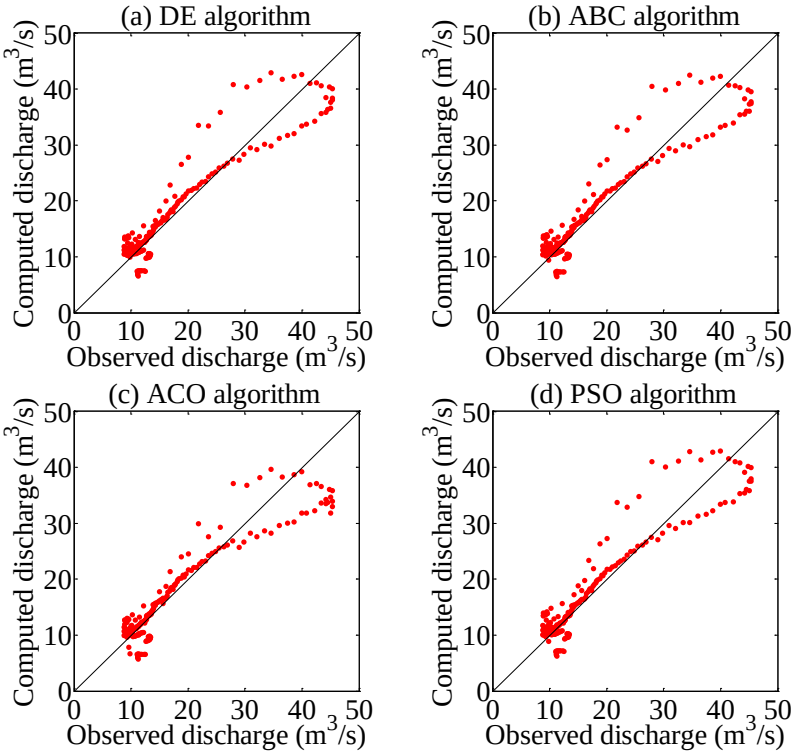

Figure 7.9 Observed and computed discharges by (a) DE (b) ABC (c) ACO and (d) PSO algorithm for the 16-hour-ahead case in the Altamaha river basin

The statistical analysis by the four algorithms is presented in Table 7.8 in order to check the significance of the differences. The ACO algorithm obtains the most approximate mean values of all data and of 95% confidence interval, which agrees with its highest ACC value in the testing period. The variances computed by the four algorithms are lower than the observations, where the ACO algorithm attains the lowest one. The reliability of the comparative study of optimization algorithms is proven by the DM test, since all the DM statistics are between -1.96 and 1.96 in Table 7.9. The preference with

regard to the DE algorithm is supported by the negative values attained between the DE and ABC/ACO algorithms. The positive DM statistic between the DE and PSO algorithms reveals lower accuracy of the former. The difference between the ABC and PSO algorithms is relatively significant with a value of 1.7325 for the DM statistic.

The training errors versus running iteration are demonstrated in Figure 7.10 for the purpose of comparing convergence speed of the optimization algorithms. Observations reveal that the DE and PSO algorithms have a similar trend to achieve convergence. The model optimized by the ACO algorithm also reaches the minimum training error before 500 running iteration, yet with slower speed. The ABC algorithm suffers the disadvantage of slow convergence speed as before, in which the training error decreases slowly and becomes constant after running 2000 times. It is noted that the ABC achieves almost the same minimum training error as the other three algorithms, which indicates that the model parameters are fully searched and optimized. This might explain the excellent performances of the ABC algorithm for the present study case when compared against the results in Figure 7.4. The running times shown in Table 7.10 imply the feature of time consuming of the ABC algorithm. The larger running iteration is not the only reason for the longer running time, since the maximum iteration by the ABC is four times larger than the DE and yet its running time is eight times longer than 0.84 hour. This means that the ABC spends more time in each iteration, and its computational efficiency is lower in comparison with the other algorithms. It can also be observed from Table 7.10 that the parameter optimization ability of the four algorithms is fairly equal for the present case, with evidence of the comparable values of recession coefficient and initial storage.

Table 7.8 Statistical analysis by various algorithms for the 16-hour-ahead case in the Altamaha river basin

Algorithm	Mean (m^3/s)	Variance (m^6/s^2)	Mean of 95% confidence interval (m^3/s)
Observed	14.5282	75.2713	12.3248
DE	15.1541	65.7083	13.0795
ABC	15.0622	64.4972	13.0153
ACO	14.3551	55.2520	12.5727
PSO	14.8639	67.4586	12.7749

Table 7.9 DM test for the forecasting accuracy between each pair of algorithms for the 16-hour-ahead case in the Altamaha river basin

Algorithms	DE	ABC	ACO	PSO
DE		-0.1490	-0.8333	1.0665
ABC	0.1490		-0.9178	1.7325
ACO	0.8333	0.9178		1.0494
PSO	-1.0665	-1.7325	-1.0494	

Table 7.10 Running time and values of special parameters by various algorithms for the 16-hour-ahead case in the Altamaha river basin

Algorithm	Maximum iteration (*MI*)	Running time (hour)	Parameters	
			recession coefficient λ	initial storage $S_{(T=0)}$
DE	500	0.84	0.4553	0.1350
ABC	2000	7.21	0.4585	0.1359
ACO	500	1.15	0.4850	0.1324
PSO	500	0.73	0.4600	0.1283

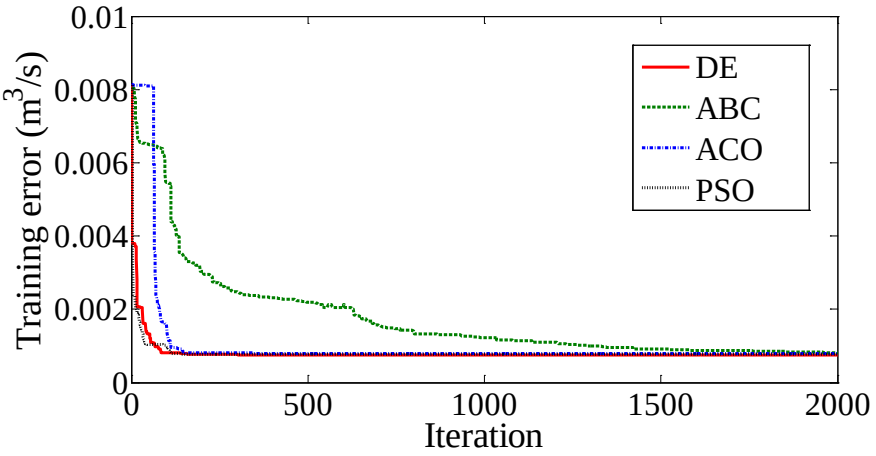

Figure 7.10 Training errors (RMSE) by various algorithms versus running iteration for the 16-hour-ahead case in the Altamaha river basin

In addition, the influences of control parameters on the DE, ABC and ACO algorithms are systematically performed. The contributions of population size to the forecasting performances are outlined in Figure 7.11 for the 16-hour-ahead forecast. The RMSE value by the DE algorithm gradually decreases before the best result is obtained with *NP*=600, and then slightly increases with the increase of *NP*. The ABC algorithm has similar tendency with the varying *NP* value and reaches the minimum RMSE value when *NP*=460. However, it is relatively sensitive to the population size when compared with the DE algorithm. The RMSE value obtained by the ACO algorithm begins with a larger one when *NP*=100 and decreases to a minimum value when *NP*=800. The three algorithms reveal comparably stable features in terms of the population size, wherein the DE and ABC algorithm are more independent.

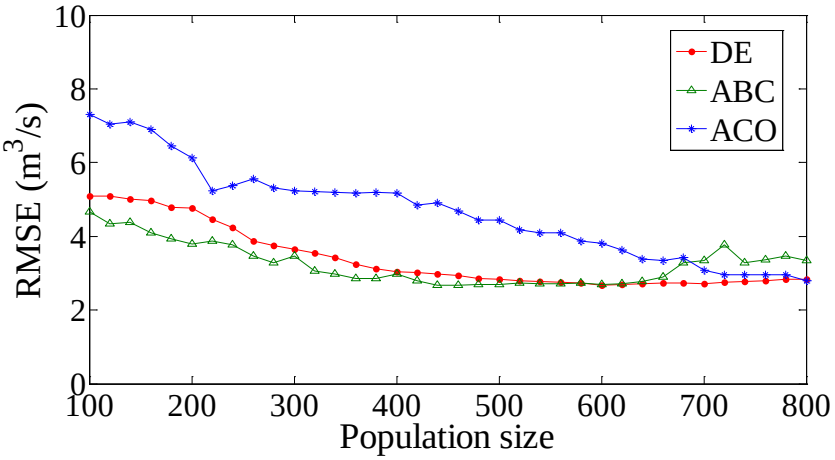

Figure 7.11 The influence of population size on forecasting performances by various algorithms for the 16-hour-ahead case in the Altamaha river basin

The performances of DE algorithm regarding two control parameters *F* and *CR* are demonstrated in Figure 7.12. The RMSE value presents comparably stable with respect to *F* in the interval of [0.2, 2] and the best result is obtained taking *F* as the value of 0.4. As observed in Figure 7.12(b), the RMSE value gradually decreases at the beginning, and drops to a small value when *CR*=0.55. Then it slightly decreases with the increase

of *CR* value. In comparison with the *F*, the DE algorithm is more sensitive to the *CR* parameter for the present case. The possible reason is that the *CR* value controls the change of the diversity of the population, thus the HNN model is more dependent on the *CR* when forecast lead time is larger. Likewise, the contribution of *Limit* parameter to the ABC algorithm is shown in Figure 7.13. The RMSE value drops sharply with increasing *Limit* firstly, and then fluctuates slightly within a small range. A minimum value is obtained when *Limit*=226 with fixed *NP*=460. The optimization becomes independent when the value of *Limit* is larger than half of the population size, which is exactly the number of employ bees. As aforementioned, the ABC algorithm would not undergo the scout bees' phase when the *Limit* exceeds the number of employed bees. The effects of different influencing parameters on the ACO algorithm are demonstrated in Figure 7.14. It is noticed that there is no regular variation for the performances with the increasing *N* value. The ACO algorithm presents stability in terms of the ρ value in view of the small change of RMSE values. The pheromone constant *Q* is highly correlated with the fitness value and controls the reinforcement process. As observed in Figure 7.14(c), the ACO is quite insensitive when the *Q* value is larger than 0.10 and the RMSE value obtained at *Q*=0 is deviant.

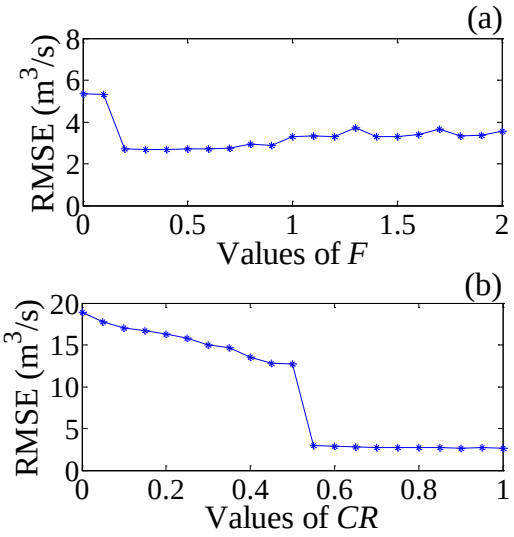

Figure 7.12 The influences of (a) *F* (b) *CR* on forecasting performances by DE algorithm for the 16-hour-ahead case in the Altamaha river basin

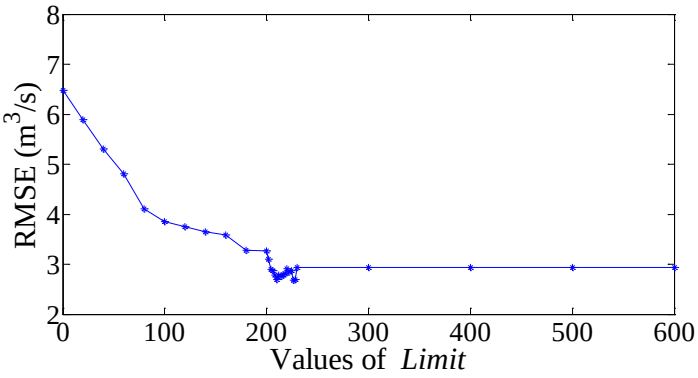

Figure 7.13 The influence of *Limit* on forecasting performances by ABC algorithm for the 16-hour-ahead case in the Altamaha river basin

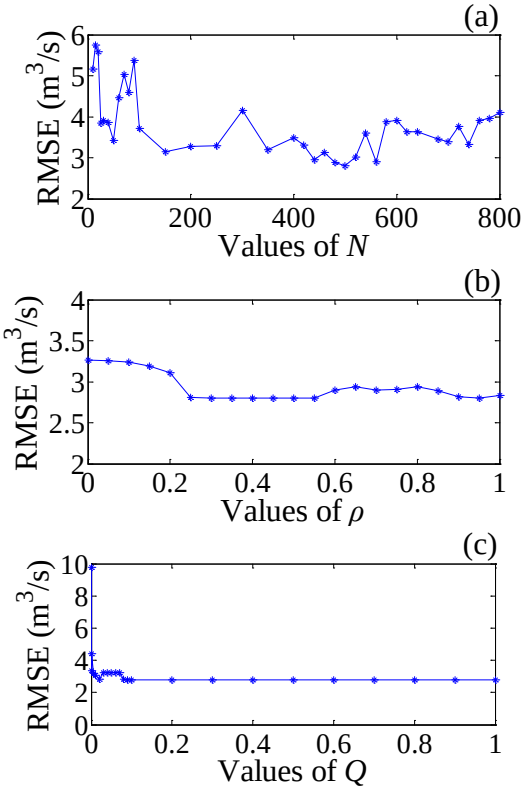

Figure 7.14 The influences of (a) N (b) ρ (c) Q on forecasting performances by ACO algorithm for the 16-hour-ahead case in the Altamaha river basin

In short, the DE, ABC and ACO algorithms present their availability in optimizing the multi-step-ahead forecasts, and show considerably equal chance of forecast by the DM test. The ABC algorithm performs excellently on the forecasting results and parameter optimization in particular, which reveals the availability in the HNN model when compared with the one-day-ahead case. However, conclusions that the ABC algorithm is better in the multi-step-ahead forecast could not be drawn since the single- and

multi-step-ahead cases adopt daily and hourly data respectively.

7.4 Applications on sediment load estimation

A further analysis on the applications of optimization algorithms on the HDFNN model for sediment load estimation is presented in this section. When compared with the HNN model, the problem of the HDFNN arises with features of more complex and discontinuous since double neural networks are employed. The stations 06088300 and 06088500 on the Muddy Creek in Montana are considered as study case sites. The same daily data are applied in the estimation model, and the best input combinations determined in Chapter 5 are adopted for single- and multi-step-ahead cases.

7.4.1 Performances on single-step-ahead estimation

It has been reported in Chapter 5 that the DE algorithm yields the best estimation results with NSEC=0.9762 for the one-day-ahead case on the Muddy Creek. The corresponding performances are summarized in Table 7.11 when optimizing the same HDFNN model by ABC, ACO and PSO algorithms. It is observed that the DE algorithm outperforms others in view of lowest RMSE and highest NSEC value in both training and testing periods. The ACO comes in second place for the generalization and estimation abilities, whose NSEC value in the testing period is 0.21% smaller than that by the DE algorithm. The ABC and PSO algorithms enjoy comparable performances during the testing period, and yet the ABC is preferred because of more fully trained. More insights on the estimation performances can be obtained in Table 7.12 through a partitioning analysis. The ACO algorithm shows its superiority in the low and medium loads particularly whereas the DE tends to better approximate the high loads. The results could be observed in Figure 7.15 by the scatter plots of the observed and estimated sediment

loads. The ABC over-estimates many observations below 500 tons/day and the PSO visibly under-estimates some dots near 1500 tons/day. The four algorithms tend to yield unsatisfactory under-estimated results since the computed dots in the lower-right corner of the figures are located far from the ideal line. The above results are obtained with *MI* =500, *NP*=600, *F*=0.4 and *CR*=1.0 in the DE algorithm; *MI* =2000, *NP*=500 and *Limit*=248 in the ABC algorithm; *MI* =500, *NP*=720, *N*=250, ρ=0.35 and *Q*=0.07 in the ACO algorithm.

Table 7.11 Training and testing performances by various algorithms in terms of RMSE and NSCE for the one-day-ahead case on the Muddy Creek

Algorithm	Training		Testing	
	RMSE (tons/day)	NSEC	RMSE (tons/day)	NSEC
DE	**284.9112**	**0.9922**	**362.8466**	**0.9762**
ABC	316.6352	0.9903	393.3081	0.9720
ACO	316.5997	0.9903	377.9417	0.9742
PSO	351.5354	0.9881	392.7150	0.9721

Table 7.12 Partitioning analysis by various algorithms for the one-day-ahead case on the Muddy Creek

Algorithm	Low load RMSE (tons/day)	Medium load RMSE (tons/day)	High load RMSE (tons/day)
DE	112.2258	161.2308	**776.1632**
ABC	142.3585	255.5686	800.1699
ACO	**71.3035**	**140.8372**	829.8670
PSO	147.0277	162.3081	833.1924

Statistical analysis is conducted in terms of the mean, variance and mean of 95% confidence interval of the observed and estimated sediment loads. As seen in Table 7.13, the four algorithms do not present significant differences with the observations, indicating their reliability in optimizing the HDFNN model. The DM test for the estimation accuracy between each pair of algorithms is provided in Table 7.14, which also proves the equality of each algorithm for the HDFNN model when the significance level of the test is supposed as 0.05. The similarity of estimation ability between ABC

and PSO algorithms is verified by the small DM statistic (i.e., 0.0431). The DE algorithm presents best accuracy in view of the negative DM statistics.

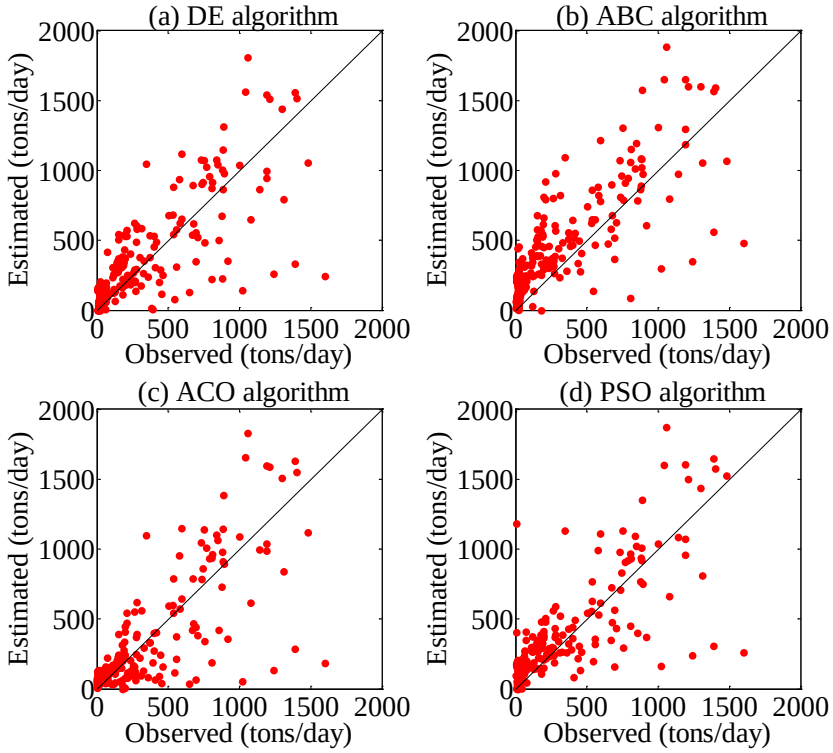

Figure 7.15 Observed and estimated sediment loads by (a) DE (b) ABC (c) ACO and (d) PSO algorithm for the one-day-ahead case on the Muddy Creek

Table 7.13 Statistical analysis by various algorithms for the one-day-ahead case on the Muddy Creek

Algorithm	Mean (tons/day)	Variance ($tons^2/day^2$)	Mean of 95% confidence interval (tons/day)
Observed	385.0736	5.5440×10^6	231.1238
DE	354.6502	5.0528×10^6	215.0894
ABC	435.1645	5.4230×10^6	293.0341
ACO	364.5258	5.1173×10^6	224.1479
PSO	362.9237	4.8201×10^6	227.3404

Table 7.14 DM test for the estimation accuracy between each pair of algorithms for the one-day-ahead case on the Muddy Creek

Algorithms	DE	ABC	ACO	PSO
DE		-1.2926	-1.4864	-1.6245
ABC	1.2926		0.7053	0.0431
ACO	1.4864	-0.7053		-0.8199
PSO	1.6245	-0.0431	0.8199	

Likewise, the convergence speed and result of the four optimization algorithms are compared. It can be noticed in Figure 7.16 that the training error optimized by each algorithm decreases to a minimum value after running 2000 times in which the one obtained by the PSO algorithm is slightly larger than the others. None of the algorithms rapidly attain its minimum training error, which is obviously larger than the others, suggesting that premature convergence does not happen. The DE and PSO have fast speed to achieve convergence while the ACO algorithm presents inefficiency at the first 200 iterations. The rate of convergence by the ABC is not as high as that by the other three algorithms, but their degrees of convergence are quite similar. Results in Table 7.15 reveal that the ABC takes almost 8 hours to achieve convergence by running 2000 iterations. The other three algorithms demonstrate comparably high computational efficiency. The values of special parameters optimized by various algorithms are provided in Table 7.15 as well. The ABC and ACO algorithms obtain approximated values of $a_{0,}\, b_0$ and initial sediment deposition SD_0^s, and the recession coefficients are respectively 0.4796 and 0.7149. The four parameters attained by the DE and PSO are quite different from those by the former two algorithms, revealing the difficulty and complexity of the HDFNN model in parameter optimization.

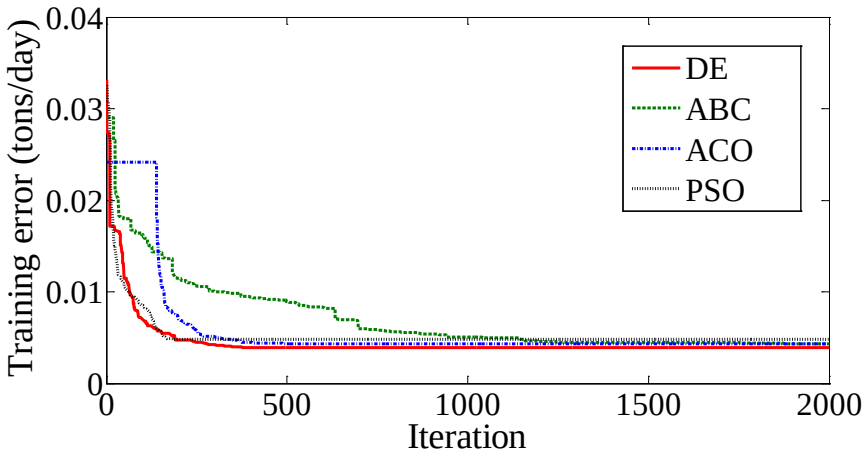

Figure 7.16 Training errors (RMSE) by various algorithms versus running iteration for the one-day-ahead case on the Muddy Creek

Table 7.15 Running time and values of special parameters by various algorithms for the one-day-ahead case on the Muddy Creek

Algorithm	Maximum iteration (*MI*)	Running time (hour)	Parameters			
			a_0	b_0	recession coefficient λ	initial storage SD_0^s
DE	500	1.32	0.0547	5.2456	0.6504	0.0091
ABC	2000	8.00	0.4794	2.3475	0.4796	0.0200
ACO	500	1.97	0.4780	2.9265	0.7149	0.0202
PSO	500	1.07	0.1643	4.3540	0.5391	0.1573

The dependence of DE, ABC and ACO algorithms on their control parameters is of concern for the HDFNN model. The influence of population size on the model performances is illustrated in Figure 7.17. The RMSE value by the DE and ABC algorithms has similar variation with increasing *NP*, and it fluctuates within a narrow range when the *NP* value falls in [220, 800]. The ACO algorithm is highly sensitive to the *NP* value, and the best result is attained when *NP*=720. This might be attributed to the mechanism of the ACO algorithm, in which the times of updating pheromone intensity depend on the *NP* value. The stability of the three algorithms in terms of other control parameters will be discussed in conjunction with the multi-step-ahead estimation in the following, since the three-day-ahead case on the Muddy Creek is considered as a study case which shares the same daily data as the present one-day-ahead case.

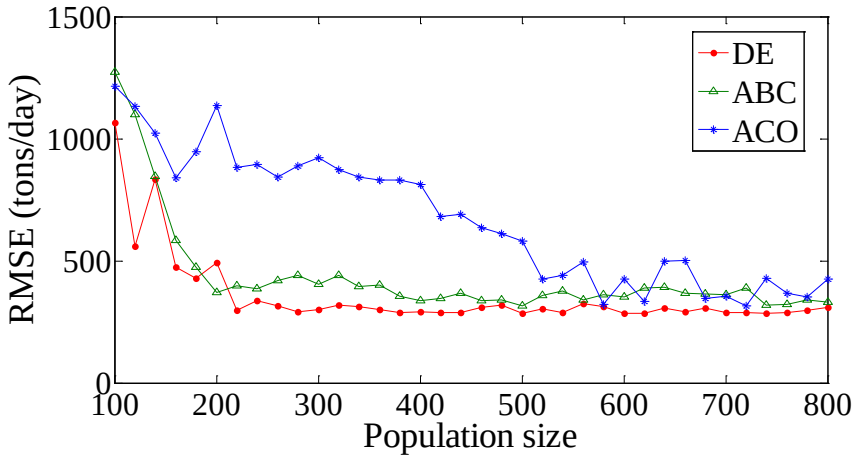

Figure 7.17 The influence of population size on estimation performances by various algorithms for the one-day-ahead case on the Muddy Creek

7.4.2 Performances on multi-step-ahead estimation

The model performances by the four optimization algorithms with respect to the three-day-ahead case on the Muddy Creek are summarized in Table 7.16. The DE outperforms others in the testing period in view of a reduction of 7.42%, 18.37% and 32.54% in RMSE value when compared with ABC, ACO and PSO algorithms respectively. It yields the best performance in the training period as well, revealing the generalization and estimation ability of the DE algorithm for the HDFNN model with large lead time. The PSO algorithm presents conflicting results between the training and testing periods, in which the model is trained excessively and yet the estimation is not as good as expected. The ABC is a preferred algorithm when compared with the ACO, in which there is a reduction of 15.55% and 11.82% in terms of RMSE value respectively in the training and testing period. The excellent performances of the ABC algorithm in the present study case disaccord with the one-day-ahead estimation, which might be due to the ability of effectively providing solutions even when the lead time is large. That is, multi-step-ahead estimations might increase dispersion and uncertainty to the optimization problem, nevertheless the operation of abandoning exhausted solutions in the ABC algorithm might avoid the possibility. Furthermore, the partitioning analysis provided in Table 7.17 reveals that the excellent performance of the DE algorithm primarily relies on its capability in accurately fitting the high loads. Its RMSE values in low and medium loads are not unsatisfactory, yet they are larger than those attained by the PSO algorithm. It is worthwhile to note that the DE algorithm consistently presents superiority in computing the high values for the sediment load. This might be attributed to the crossover process of the DE algorithm in which the diversity of the population is satisfied. On the other hand, the high RMSE value in high loads seems to be the reason for the poor performances of the PSO algorithm. Despite that, the PSO is an appropriate

algorithm for the HDFNN model the same as the other algorithms, since the NSEC value obtained is as high as 0.9360. The scatter plots of observed and estimated sediment loads by the four algorithms are demonstrated in Figure 7.18. The ABC and PSO algorithms improperly under-estimate some dots as near-zero values. The DE algorithm can fit the observations without too much over-estimation while the ACO presents excellent performances instead of unduly under-estimating the observations.

Table 7.16 Training and testing performances by various algorithms in terms of RMSE and NSCE for the three-day-ahead case on the Muddy Creek

Algorithm	Training		Testing	
	RMSE (tons/day)	NSEC	RMSE (tons/day)	NSEC
DE	**274.5113**	**0.9927**	**402.3409**	**0.9709**
ABC	352.6813	0.9880	434.6173	0.9658
ACO	417.6449	0.9832	492.8989	0.9563
PSO	307.0586	0.9909	596.3998	0.9360

Table 7.17 Partitioning analysis by various algorithms for the three-day-ahead case on the Muddy Creek

Algorithm	Low load RMSE (tons/day)	Medium load RMSE (tons/day)	High load RMSE (tons/day)
DE	101.8828	169.0317	**868.8377**
ABC	97.3824	217.8088	933.3839
ACO	186.6741	302.9915	1004.208
PSO	**97.0076**	**148.0918**	1325.044

In addition, the significance of the differences of the four algorithms is investigated in terms of statistical analysis and DM test, the results of which are presented in Tables 7.18 and 7.19. As can be seen, the mean values of total and 95% confidence interval by the ACO algorithm are not close to the observations most. The PSO algorithm obtains a discrete variance value when compared with the others. Thus the ACO and PSO algorithms seem to have inferior performances in estimation accuracy. This conclusion is further substantiated by the positive DM statistics attained between ACO/PSO and the other two algorithms. Despite that, the equal chance of estimation accuracy of the four algorithms is confirmed since all the DM statistics outlined in Table 7.19 are within

the range of -1.96 and 1.96. The priority of the DE algorithm is verified by the negative DM statistics between DE and the other three. The PSO yields worse accuracy than the ACO algorithm with DM statistic=0.6230, which is highly consistent with results in Table 7.16. The above results are on the basis of computation with *MI* =500, *NP*=520, *F*=0.4 and *CR*=1.0 in the DE algorithm; *MI* =2000, *NP*=500 and *Limit*=244 in the ABC algorithm; *MI* =500, *NP*=760, *N*=400, ρ=0.35 and *Q*=0.07 in the ACO algorithm.

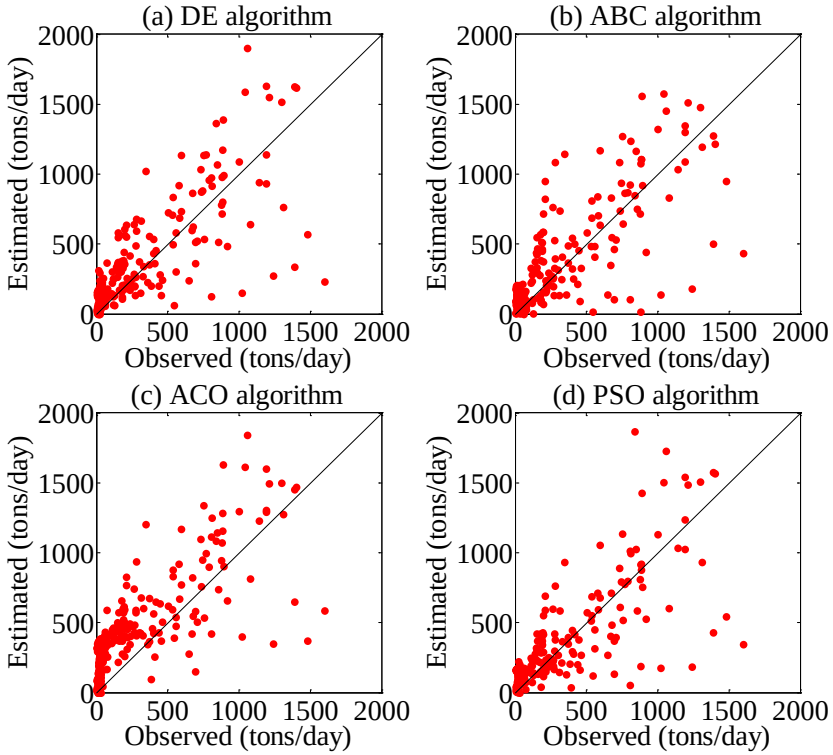

Figure 7.18 Observed and estimated sediment loads by (a) DE (b) ABC (c) ACO and (d) PSO algorithm for the three-day-ahead case on the Muddy Creek

Table 7.18 Statistical analysis by various algorithms for the three-day-ahead case on the Muddy Creek

Algorithm	Mean (tons/day)	Variance (m^6/s^2)	Mean of 95% confidence interval (tons/day)
Observed	387.1635	5.5740×10^6	232.3700
DE	363.6521	4.7464×10^6	228.0751
ABC	381.3827	4.4618×10^6	251.3216
ACO	464.1081	4.6973×10^6	334.2639
PSO	340.3519	3.4365×10^6	224.7561

Table 7.19 DM test for the estimation accuracy between each pair of algorithms for the three-day-ahead case on the Muddy Creek

Algorithms	DE	ABC	ACO	PSO
DE		-0.9092	-1.1528	-1.0940
ABC	0.9092		-1.0739	-1.0459
ACO	1.1528	1.0739		-0.6230
PSO	1.0940	1.0459	0.6230	

When examining the convergence of the four algorithms, the results of training error versus running iteration are illustrated in Figure 7.19. It is observed that the ACO algorithm yields a relatively larger training error after running 2000 times. The performances of the training errors produced by the four algorithms are in agreement with the training performances provided in Table 7.16. The convergence speed is another index of training ability, revealing the computational efficiency of optimization algorithms. The PSO, DE, ACO and ABC are accordingly placed in order in terms of fast convergence speed for the present case. Likewise, the running times as well as the values of optimization parameters are presented in Table 7.20. The ABC algorithm presents feature of most time consumption, however, has comparable ability in finding the parameters with the PSO algorithm. The DE algorithm obtains quite different parameters, and they might be the most adaptive ones since they are accompanied by the best estimation performances.

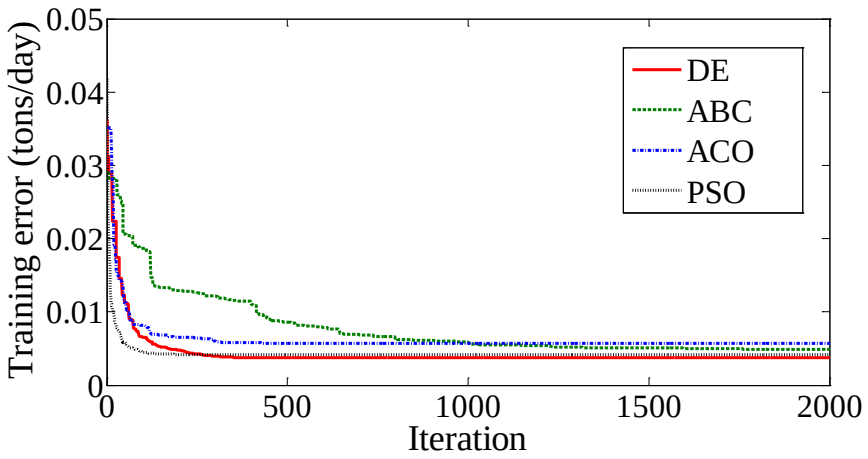

Figure 7.19 Training errors (RMSE) by various algorithms versus running iteration for the three-day-ahead case on the Muddy Creek

Table 7.20 Running time and values of special parameters by various algorithms for the three-day-ahead case on the Muddy Creek

Algorithm	Maximum iteration (*MI*)	Running time (hour)	Parameters			
			a_0	b_0	recession coefficient λ	initial storage SD_0^s
DE	500	1.13	0.3162	3.4912	0.7238	0.0412
ABC	2000	8.44	0.1993	5.0023	0.4789	0.0024
ACO	500	2.08	0.0352	4.7520	0.5514	0.1079
PSO	500	1.03	0.1977	5.3080	0.2612	0.0656

The stability of DE, ABC and ACO algorithms in terms of the population size *NP* is outlined in Figure 7.20. Similar to the one-day-ahead case, the ACO is relatively sensitive to the *NP* value in comparison with the other two algorithms. The influences of *F* and *CR* of the DE algorithm on the model performances are demonstrated respectively for the one- and three-day-ahead cases in Figure 7.21. They enjoy similar tendency with increasing *F* and *CR*, and both obtain best results when *F*=0.4 and *CR*=1.0. When compared with the *CR* value, the *F* seems to have more influences on the model performances. This might be a common specialty for the DE algorithm. Likewise, the sensitivity of ABC algorithm to the *Limit* value is depicted in Figure 7.22. The RMSE values gradually decrease when *Limit* increases, and fluctuate within a small range when the *Limit* value approaches the half of the population size, both for the one- and three-day-ahead estimations. This fairly agrees with the cases for the river flow forecasting, where the ABC algorithm is quite dependent of the *Limit* parameter. Details regarding the control parameters of the ACO algorithm are shown in Figure 7.23. The one- and three-day-ahead estimations still enjoy similar performances with the changes of *N*, ρ and *Q*. The RMSE decreases rapidly at the beginning and becomes relatively stable with the increase of *N* value. In addition, the ACO algorithm appears to be insensitive to the ρ value within the range of [0.1, 0.8]. The minimum RMSE values are attained with *Q*=0.07, and the model performances are independence of the *Q* value after the best results are achieved.

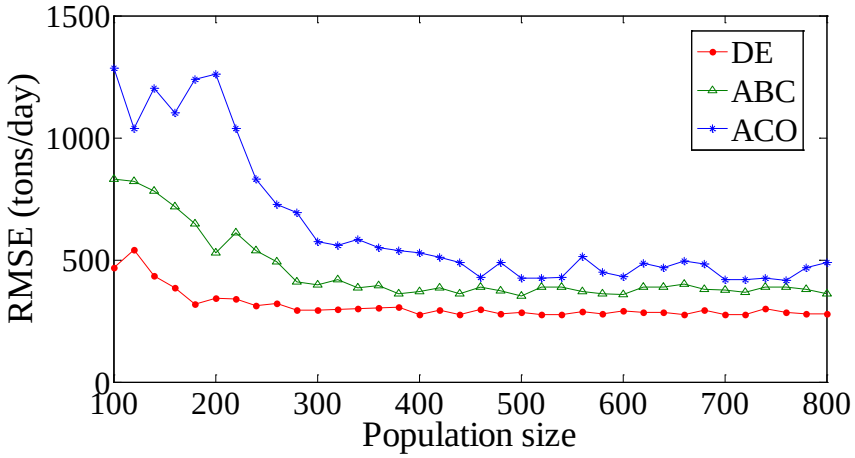

Figure 7.20 The influence of population size on estimation performances by various algorithms for the three-day-ahead case on the Muddy Creek

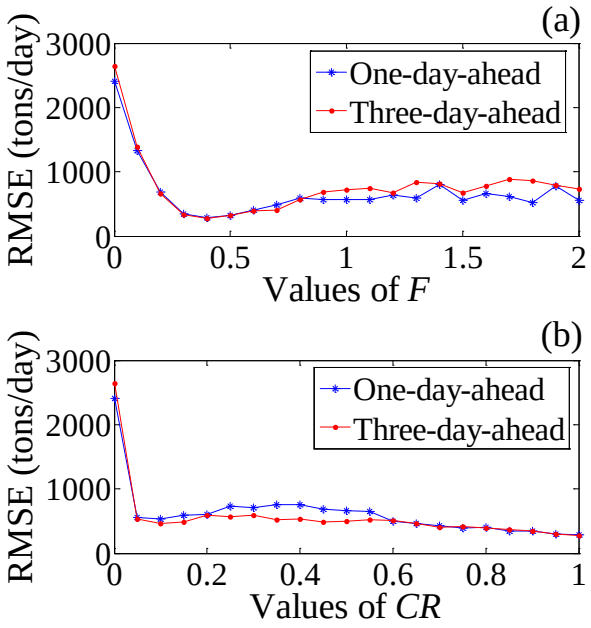

Figure 7.21 The influences of (a) *F* (b) *CR* on estimation performances by DE algorithm for the cases on the Muddy Creek

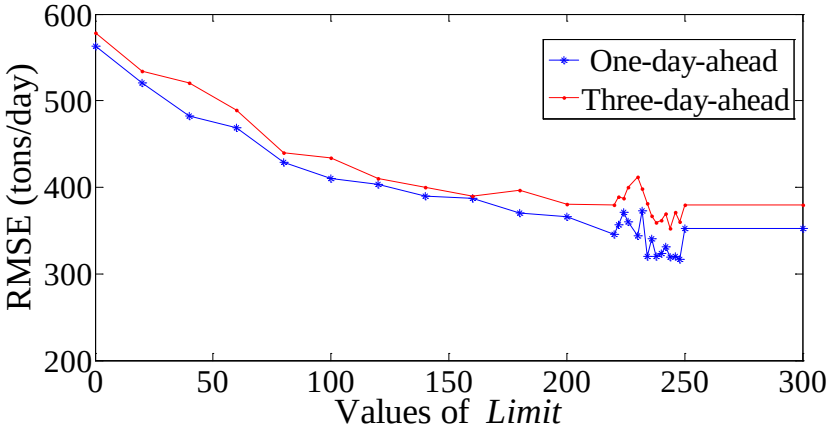

Figure 7.22 The influence of *Limit* on estimation performances by ABC algorithm for the cases on the Muddy Creek

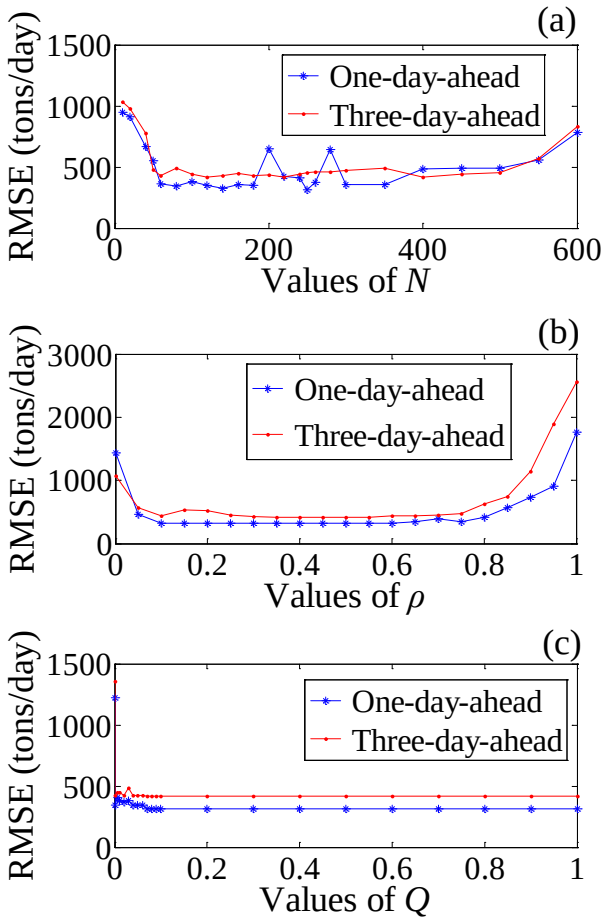

Figure 7.23 The influences of (a) N (b) ρ (c) Q on estimation performances by ACO algorithm for the cases on the Muddy Creek

7.5 Summary

The applications of population-based optimization algorithms in hydrology community have been a great interest for decades. However, there is a lack of systematical investigation on the performances of the algorithms coupled with neural network models. In this chapter, the performances of three population-based optimization algorithms, namely DE, ABC and ACO, are compared when evolving the HNN and HDFNN models. The computation efficiency and parameter optimization is examined as well. Besides, the effects of control parameters on each algorithm are quantified additionally to improve the understanding about the optimization process. Thus this study attempts to determine a relatively reliable and stable algorithm for the nonlinear

and non-differentiable optimizations. From the results on the HNN and HDFNN models, following conclusions are drawn:

(a) The chance of accuracy of the DE, ABC and ACO algorithms in optimizing the HNN and HDFNN models is comparably equal by the DM test when the significance level is assigned as 0.05. The reliability of the present comparative study on the optimization algorithms is confirmed accordingly.

(b) The DE is found to be a more appropriate algorithm in terms of generalization and estimation. Particularly, the high values of the HNN and HDFNN models are consistently best fitted by the DE algorithm. The crossover process aiming at providing diversity of the population might be a possible reason.

(c) The ABC algorithm appears to be more capable in optimizing the multi-step-ahead cases. This may be explained by the fact that the solutions of the ABC algorithm are effectively provided without exhausted ones even when the cases are dispersed and uncertain with large lead time. On the other hand, the ABC algorithm presents computational inefficiency since it achieves the convergence slowly.

(d) The results of parameter optimizations in the HDFNN model are more dependent on the algorithms when compared with those in the HNN model because of the more complicated and discontinuous problems in the sediment load estimation.

(e) The DE and ABC algorithms are more adaptive than the ACO with the population size. The dependence of ACO algorithm on the *NP* value is attributed to the mechanism that *NP* specifies the times of updating pheromone intensity. Besides, the ACO is inferior to the other two algorithms with the noteworthy drawback of employing more control parameters.

(f) The dependences of control parameters on each algorithm are generally quite similar in single- and multi-step-ahead cases. In particular, the DE algorithm is more sensitive to the *CR* in comparison with the *F* parameter when lead time is larger.

The probable reason is that the *CR* value controlling the change of the diversity of the population is more influencing in multi-step-ahead cases.

8 Conclusions and Future Work

8.1 Conclusions

River flow forecasting and sediment load estimation have been recognized as significant issues in the hydrological field, and customarily undertaken by data-driven models. Despite the considerable amount of research on these models, most of them are totally a “black box” which could not reflect the physical relations between the input and output variables. Besides, these models tend to be processed with inappropriate input data and over-simplified assumptions, thus coupled with substantial uncertainty. The present thesis has advanced flexible hybrid models in which physical activation functions are introduced into the neural networks for river flow and sediment load. In this regards, the highly nonlinear, time-varying and fuzzy features of flow and sediment data are taken into account. Additionally, the determination of input variables, uncertainty analysis of proposed models and optimization algorithms employed in the calibration have been included in this study.

The thesis first proposes a novel hybrid neural network (HNN) model by integrating with the fuzzy pattern-recognition concept and continuity equation, which work as activation functions in the hidden and output layers directly. The model is therefore, capable of reflecting fuzzy and time-varying features of river flows, by classifying inputs into different patterns and assuming the downstream station as a nonlinear storage reservoir. Three benchmarking models (namely ANN, FPNN and HYMN) are employed and two study cases are investigated under both daily and hourly scenarios. With respect to forecasting high flow values, the HNN model is identified as the

preferred tool which mainly benefits from the introduction of fuzzy pattern-recognition. It appropriately fits the total observations with the best accuracy since the continuity equation produces consistent computation. As forecasting lead time increases, the performance of the HNN model does not markedly deteriorate. Further, the superiority of HNN model over the other three is more salient in the case with larger storage capacity of the reservoir, particularly from the perspective of multi-step-ahead forecasting.

A hybrid double feedforward neural network (HDFNN) model is then developed for suspended sediment load (*SSL*), by combining fuzzy pattern-recognition and continuity equation into a structure of double neural networks. The upstream sediment and flow inputs solely contribute to the output in terms of double neural network, since the upstream sediment is not a connatural variable with the flows and can not be employed in the pattern classification. The estimates based on HDFNN models are compared with three benchmark models (namely MFNN, DPFNN and HFNN) using daily dataset in two study cases. The generalization and estimation abilities of HDFNN model are attested with the lowest RMSE and highest NSEC values. Besides, it can reproduce medium and high loads appropriately as indicated by the partitioning analysis. The HDFNN model is also observed to present excellent performances in multi-step-ahead estimations, particularly in high loads, owing to the effective displayed storage capacity of the reservoir with large lead time. The case with a larger recession coefficient renders greater advantage of the HDFNN model over the others as well.

The request of determining the best input variables for the HNN and HDFNN models has been processed. River flow and precipitation data from the upstream sections are imposed as potential inputs for downstream river flow. It is found that the efficiency of

the input variables depends on the basin boundary. Areal precipitation is an appropriate input variable coupled with all observed upstream flows in general. The upstream flows may be redundant and uncertain with increasing lead time. When the studied river basin is small, the contribution of upstream flow inputs to the output seems to be insensitive and areal precipitation may not be an acting input variable. Observations on the hourly forecasts reveal that certain input variables, including the precipitation may be uncertain and unstable for the HNN model when forecast lead time is shorter than the flow travel time. For the sediment load estimation, river flows from the upstream and downstream stations with different ahead of times are selected to formulate possible input combinations. The downstream sediment seems to be more sensitive to upstream flows when the studied area is small. The downstream flow is a critical factor affecting high sediment loads substantially. With the increase of lead time, the best performances on low, medium and high loads tend to be attained from the different input combinations, revealing the fuzzy feature of the sediment loads.

The reliability of the proposed models is evaluated by uncertainty analysis, so as to attain interval predictions of the flow and sediment. The LUBE is a straight-forward method that can construct neural network based models with two output neurons and directly approximate the lower and upper bounds of PIs. Applications on the river flow forecasting models reveal the viability of HNN model from the perspective of one-day-ahead and 16-hour-ahead cases with 90% and 95% confidence level. In regard to the 99% CL case, the HNN model generates valid and narrow PIs with a lead time of one day, although the improvements over the other three models are not as significant. The large lead time in the 16-hour-ahead case increases the uncertainties in the inputs, and thus leads to failure to ensure the required coverage probability with confidence level 99%. As for the sediment load estimation, the superiority degree of the HDFNN

over its counterparts does not significantly mitigate with the increase of lead time for the 90% and 95% CL cases. The more compact structure of HDFNN model renders narrower PIs entailing less uncertainty. Further, it could produce better lower bounds of PIs owing to the integration of the continuity equation and the employment of the recession coefficients. Partitioning analysis reveals that the HDFNN consistently constructs narrow PIs with 100% enclosed observations in low and medium loads, although with invalid PIs in high loads with 95% CL.

The performances of three population-based optimization algorithms, namely differential evolution (DE), artificial bee colony (ABC) and ant colony optimization (ACO) for evolving the HNN and HDFNN models are compared. It is firstly indicated by the DM test that the chance of accuracy of the three algorithms in optimizing the HNN and HDFNN models is comparably equal when the significance level is assigned as 0.05. The DE is found to be a more appropriate algorithm in terms of generalization and estimation, particularly for fitting the high values of the two models. The ABC algorithm appears to be more capable in optimizing the multi-step-ahead cases, but on the other hand, presents computational inefficiency. As far as the stability of the algorithms is concerned, the DE and ABC algorithms are more adaptive than the ACO with the population size. Besides, the ACO is inferior to the other two algorithms with the noteworthy drawback of employing more control parameters.

One of the major contributions of this research is to explore hybrid neural network models with conceptual activation functions for river flow and sediment load. The effectiveness and reliability of the proposed HNN and HDFNN models have been verified respectively by the point and interval predictions, involving single- and multi-step-ahead cases. The LUBE method has proven as a promising technique

coupling with hybrid neural network models for uncertainty analysis. In addition, the DE algorithm has been identified as an appropriate optimization method in solving the highly nonlinear, non-differential and time-varying problems of the proposed models.

8.2 Suggestions and future work

More studies as an attempt to improve the accuracy of hydrological predictions and elucidate physical mechanisms in data-driven models can be explored in the future. The followings are the suggestions for research.

Firstly, conceptual principles such as the continuity equations of momentum and energy are satisfied in a river system, which are worth being considered in the river flow and sediment modeling in addition to the mass conservation. The major challenge is how to directly increase the physical-plausibility of the data-driven models, in which integrating conceptual activation functions in neural networks has already been implemented in this study. Furthermore, the present work is the first application of considering the physics embedded with the structure of double neural network for modeling suspended sediment. This idea can be referred and extended for other hydrological models when the input variables have different effects on the output and suggested to be separately imported to the models.

Secondly, it can be observed that the peak values of the flow and sediment are generally poorly fitted although the HNN and HDFNN models yield the best performances in comparison with their counterpart models. New methods that could better capture the fuzzy feature of the flow and sediment and improve the prediction of peak values are suggested to be developed in future. In order to explore the potential of physical

input-output relation, it is strongly recommended that data preprocessing techniques be employed to reduce redundant inputs and improve the mapping ability. The partitioning analysis in the sediment estimation encourages the employment of modular approaches, which could produce the sediment load separately in terms of low, medium and high loads.

Third, the current study does not take into account the rainfall as an input variable on the sediment load, which may be as important in the river flow forecasting applications. Also owing to the scarcity of relevant data, the spatial variability of the flow and sediment inputs on the downstream sediment load has not been sufficiently examined. The hourly sample sediment estimation should be undertaken as well to ensure a wider application of the HDFNN model.

Finally, the LUBE is found to be effective in this study using DE algorithm to minimize the objective function. Further research is recommended to focus on the other optimization algorithms in addressing the highly nonlinear, discontinuous and non-differentiable problem in LUBE method. In addition, the identification of optimal input variables for the interval predictions should be undertaken using appropriate input selection techniques.

References

Abrahart, R. J. (2003). Neural network rainfall-runoff forecasting based on continuous resampling. Journal of Hydroinformatics, 5(1), 51-61.

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

Abrahart, R. J., See, L. and Kneale, P. E. (2001). Investigating the role of saliency analysis with a neural network rainfall-runoff model. Computers & Geosciences, 27(8), 921-928.

Adamowski, J., Fung Chan, H., Prasher, S. O., Ozga-Zielinski, B. and Sliusarieva, A. (2012). Comparison of multiple linear and nonlinear regression, autoregressive integrated moving average, artificial neural network, and wavelet artificial neural network methods for urban water demand forecasting in Montreal, Canada. Water Resources Research, 48(1).

Adeli, H. and Hung, S. L. (1994). Machine learning: neural networks, genetic algorithms, and fuzzy systems: John Wiley & Sons, Inc.

Adib, A. and Jahanbakhshan, H. (2013). Stochastic approach to determination of suspended sediment concentration in tidal rivers by artificial neural network and genetic algorithm. Canadian Journal of Civil Engineering, 40(4), 299-312.

Adib, A. and Mahmoodi, A. (2017). Prediction of suspended sediment load using ANN GA conjunction model with Markov chain approach at flood conditions. Ksce Journal of Civil Engineering, 21(1), 447-457.

Afan, H. A., El-shafie, A., Mohtar, W. and Yaseen, Z. M. (2016). Past, present and prospect of an Artificial Intelligence (AI) based model for sediment transport prediction. Journal of Hydrology, 541, 902-913.

Afan, H. A., El-Shafie, A., Yaseen, Z. M., Hameed, M. M., Mohtar, W. and Hussain, A. (2015). ANN based sediment prediction model utilizing different input scenarios. Water Resources Management, 29(4), 1231-1245.

Afshar, A., Sharifi, F. and Jalali, M. R. (2009). Non-dominated archiving multi-colony ant algorithm for multi-objective optimization: Application to multi-purpose reservoir operation. Engineering Optimization, 41(4), 313-325.

Afshar, M. H. (2009). Elitist mutated particle swarm optimisation algorithms: application to reservoir operation problems. Proceedings of the Institution of Civil Engineers-Water Management, 162(6), 409-417.

Agrawal, S. K. and Sahu, O. P. (2015). Artificial bee colony algorithm to design two-channel quadrature mirror filter banks. Swarm and Evolutionary Computation, 21, 24-31.

Ahmad, S. and Simonovic, S. P. (2005). An artificial neural network model for generating hydrograph from hydro-meteorological parameters. Journal of Hydrology, 315(1), 236-251.

Akhtar, M. K., Corzo, G. A., van Andel, S. J. and Jonoski, A. (2009). River flow forecasting with artificial neural networks using satellite observed precipitation pre-processed with flow length and travel time information: case study of the Ganges river basin. Hydrology and Earth System Sciences, 13(9), 1607-1618.

Alizadeh, M. J., Kavianpour, M. R., Kisi, Ö. and Nourani, V. (2017). A new approach for simulating and forecasting the rainfall-runoff process within the next two months. Journal of Hydrology, 548, 588-597.

Alp, M. and Cigizoglu, H. K. (2007). Suspended sediment load simulation by two artificial neural network methods using hydrometeorological data. Environmental Modelling & Software, 22(1), 2-13.

Altunkaynak, A. (2010). Suspended sediment concentration prediction by Geno-Kalman filtering. Expert Systems with Applications, 37(12), 8583-8589.

Alvisi, S., Bernini, A. and Franchini, M. (2013). A conceptual grey rainfall-runoff model for simulation with uncertainty. Journal of Hydroinformatics, 15(1), 1-20.

Alvisi, S., Creaco, E. and Franchini, M. (2012). Crisp discharge forecasts and grey uncertainty bands using data-driven models. Hydrology Research, 43(5), 589.

Alvisi, S. and Franchini, M. (2011). Fuzzy neural networks for water level and discharge forecasting with uncertainty. Environmental Modelling & Software, 26(4), 523-537.

Alvisi, S. and Franchini, M. (2012). Grey neural networks for river stage forecasting with uncertainty. Physics and Chemistry of the Earth, Parts A/B/C, 42, 108-118.

Amemiya, T. (1983). Nonlinear regression models. Handbook of econometrics, 1, 333-389.

Amiri, E. (2015). Forecasting daily river flows using nonlinear time series models. Journal of Hydrology, 527, 1054-1072.

Anctil, F., Perrin, C. and Andreassian, V. (2004). Impact of the length of observed records on the performance of ANN and of conceptual parsimonious rainfall-runoff forecasting models. Environmental Modelling & Software, 19(4), 357-368.

Anderson, M. G. and Bates, P. D. (1994). Initial testing of a two-dimensional finite element model for floodplain inundation. Paper presented at the Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences.

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), 22-34.

Arabas, J. (1994). A genetic approach to the hopfield neural-network in the optimization problems. Bulletin of the Polish Academy of Sciences-Chemistry, 42(1), 59-66.

Araujo, P., Astray, G., Ferrerio-Lage, J. A., Mejuto, J. C., Rodriguez-Suarez, J. A. and Soto, B. (2011). Multilayer perceptron neural network for flow prediction. Journal of Environmental Monitoring, 13(1), 35-41.

Ardiclioglu, M., Kisi, Ö. and Haktanir, T. (2007). Suspended sediment prediction using two different feed-forward back-propagation algorithms. Canadian Journal of Civil Engineering, 34(1), 120-125.

Aronica, G., Bates, P. and Horritt, M. (2002). Assessing the uncertainty in distributed model predictions using observed binary pattern information within GLUE. Hydrological Processes, 16(10), 2001-2016.

Asadnia, M., Chua, L. H. C., Qin, X. S. and Talei, A. (2014). Improved particle swarm optimization-based artificial neural network for rainfall-runoff modeling. Journal of Hydrologic Engineering, 19(7), 1320-1329.

Ashena, R. and Moghadasi, J. (2011). Bottom hole pressure estimation using evolved neural networks by real coded ant colony optimization and genetic algorithm. Journal of Petroleum Science and Engineering, 77(3-4), 375-385.

Asselman, N. E. M. (2000). Fitting and interpretation of sediment rating curves. Journal of Hydrology, 234(3-4), 228-248.

Atkinson, S. E., Sivapalan, M., Woods, R. A. and Viney, N. R. (2003). Dominant physical controls on hourly flow predictions and the role of spatial variability: Mahurangi catchment, New Zealand. Advances in Water Resources, 26(3), 219-235.

Awan, S. M., Aslam, M., Khan, Z. A. and Saeed, H. (2014). An efficient model based on artificial bee colony optimization algorithm with Neural Networks for electric load forecasting. Neural Computing & Applications, 25(7-8), 1967-1978.

Aytek, A. and Kisi, Ö. (2008). A genetic programming approach to suspended sediment modelling. Journal of Hydrology, 351(3), 288-298.

Azamathulla, H. M., Cuan, Y. C., Ab Ghani, A. and Chang, C. K. (2013). Suspended sediment load prediction of river systems: GEP approach. Arabian Journal of Geosciences, 6(9), 3469-3480.

Babu, B. V. and Angira, R. (2003). Optimization of water pumping system using differential evolution strategies. Paper presented at the The Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems, Singapore.

Badrzadeh, H., Sarukkalige, R. and Jayawardena, A. W. (2015). Hourly runoff forecasting for flood risk management: Application of various computational intelligence models. Journal of Hydrology, 529, 1633-1643.

Bandurski, K. and Kwedlo, W. (2010). A Lamarckian hybrid of differential evolution and conjugate gradients for neural network training. Neural Processing Letters, 32(1), 31-44.

Bao, A. M., Liu, H. L., Chen, X. and Pan, X. L. (2011). The effect of estimating areal rainfall using self-similarity topography method on the simulation accuracy of runoff

prediction. Hydrological Processes, 25(22), 3506-3512.

Basu, M. (2013). Artificial bee colony optimization for multi-area economic dispatch. International Journal of Electrical Power & Energy Systems, 49, 181-187.

Benítez, J. M., Castro, J. L. and Requena, I. (1997). Are artificial neural networks black boxes? Neural Networks, IEEE Transactions on, 8(5), 1156-1164.

Beven, K. and Binley, A. (1992). The future of distributed models: model calibration and uncertainty prediction. Hydrological Processes, 6(3), 279-298.

Beven, K., Lamb, R., Quinn, P., Romanowicz, R., Freer, J. and Singh, V. (1995). Topmodel. Computer models of watershed hydrology., 627-668.

Bland, J. A. (2001). Optimal structural design by ant colony optimization. Engineering Optimization, 33(4), 425-443.

Blasone, R. S., Vrugt, J. A., Madsen, H., Rosbjerg, D., Robinson, B. A. and Zyvoloski, G. A. (2008). Generalized likelihood uncertainty estimation (GLUE) using adaptive Markov Chain Monte Carlo sampling. Advances in Water Resources, 31(4), 630-648.

Blum, C., Chiong, R., Clerc, M., De Jong, K., Michalewicz, Z., Neri, F. and Weise, T. (2012). "Evolutionary optimization" in: Variants of evolutionary algorithms for real-world applications. Chiong, R., Weise, T., Michalewicz, Z. (Eds). Springer, 2012, pp. 1-29.

Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys (CSUR), 35(3), 268-308.

Boudjelaba, K., Ros, F. and Chikouche, D. (2014). An efficient hybrid genetic algorithm to design finite impulse response filters. Expert Systems with Applications, 41(13), 5917-5937.

Boukhrissa, Z., Khanchoul, K., Le Bissonnais, Y. and Tourki, M. (2013). Prediction of sediment load by sediment rating curve and neural network (ANN) in El Kebir catchment, Algeria. Journal of earth system science, 122(5), 1303-1312.

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(1), 75-92.

Breinholt, A., Moller, J. K., Madsen, H. and Mikkelsen, P. S. (2012). A formal statistical approach to representing uncertainty in rainfall-runoff modelling with focus on residual analysis and probabilistic output evaluation - distinguishing simulation and prediction. Journal of Hydrology, 472, 36-52.

Brezkova, L., Stary, M. and Dolezal, P. (2010). The real-time stochastic flow forecast. Soil and Water Research, 5(2), 49-57.

Browning, L. S., Bauder, J. W., Hershberger, K. E. and Sessoms, H. (2005). Irrigation return flow sourcing of sediment and flow augmentation in receiving streams: A case study. Journal of Soil and Water Conservation, 60(3), 134-141.

Bullinaria, J. A. and AlYahya, K. (2014). Artificial bee colony training of neural networks: comparison with back-propagation. Memetic Computing, 6(3), 171-182.

Buyukyildiz, M., Tezel, G. and Yilmaz, V. (2014). Estimation of the change in lake water level by artificial intelligence methods. Water Resources Management, 28(13), 4747-4763.

Campolo, M., Andreussi, P. and Soldati, A. (1999). River flood forecasting with a neural network model. Water Resources Research, 35(4), 1191-1197.

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, Parts A/B/C, 31(18), 1164-1171.

Chang, F. J., Chen, P. A., Lu, Y. R., Huang, E. and Chang, K. Y. (2014). Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. Journal of Hydrology, 517, 836-846.

Chang, T. K., Talei, A., Alaghmand, S. and Ooi, M. P. L. (2017). Choice of rainfall inputs for event-based rainfall-runoff modeling in a catchment with multiple rainfall stations using data-driven techniques. Journal of Hydrology, 545, 100-108.

Chau, K. W. (2006). Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. Journal of Hydrology, 329(3), 363-367.

Chau, K. W. (2007). A split-step particle swarm optimization algorithm in river stage forecasting. Journal of Hydrology, 346(3-4), 131-135.

Chau, K. W. and Cheng, C. T. (2002). Real-time prediction of water stage with artificial neural network approach AI 2002: Advances in Artificial Intelligence (pp. 715-715): Springer.

Chen, C. H., Lin, C. J. and Lin, C. T. (2009). Nonlinear system control using adaptive neuralfuzzy networks based on a modified differential evolution. IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews, 39(4), 459-473.

Chen, J. J., Zheng, J. H., Wu, P., Zhang, L. L. and Wu, Q. H. (2017). Dynamic particle swarm optimizer with escaping prey for solving constrained non-convex and piecewise optimization problems. Expert Systems with Applications, 86, 208-223.

Chen, J. Y., Solomatine, D. P. and Xu, Y. (2015). Uncertainty prediction in hydrological modelling: case of Dapoling-Wangjiaba catchment in Huai River Basin. Proceedings of the 36th Iahr World Congress: Deltas of the Future and What Happens Upstream, 5640-5646.

Chen, L., Ye, L., Singh, V., Zhou, J. Z. and Guo, S. L. (2014). Determination of input for artificial neural networks for flood forecasting using the copula entropy method. Journal of Hydrologic Engineering, 19(11).

Chen, S., Shao, D. G., Li, X. D. and Lei, C. X. (2016). Simulation-optimization modeling of conjunctive operation of reservoirs and ponds for irrigation of multiple crops using an improved artificial bee colony algorithm. Water Resources Management, 30(9), 2887-2905.

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(7), 1525-1540.

Chen, Y. H. and Chang, F. J. (2009). Evolutionary artificial neural networks for hydrological systems forecasting. Journal of Hydrology, 367(1-2), 125-137.

Cheng, C.-T., Lin, J.-Y., Sun, Y.-G. and Chau, K. (2005). Long-term prediction of discharges in Manwan Hydropower using adaptive-network-based fuzzy inference systems models Advances in natural computation (pp. 1152-1161): Springer.

Chiong, R., Weise, T. and Michalewicz, Z. (2012). Variants of evolutionary algorithms for real-world applications: Springer.

Chitsaz, N., Azarnivand, A. and Araghinejad, S. (2016). Pre-processing of data-driven river flow forecasting models by singular value decomposition (SVD) technique. Hydrological Sciences Journal-Journal des Sciences Hydrologiques, 61(12), 2164-2178.

Choi, H. T. and Beven, K. (2007). Multi-period and multi-criteria model conditioning to reduce prediction uncertainty in an application of TOPMODEL within the GLUE framework. Journal of Hydrology, 332(3-4), 316-336.

Choi, S. U., Park, M. and Kang, H. (2007). Numerical simulations of cellular secondary currents and suspended sediment transport in open-channel flows over smooth-rough bed strips. Journal of Hydraulic Research, 45(6), 829-840.

Chryssolouris, G., Lee, M. and Ramsey, A. (1996). Confidence interval prediction for neural network models. IEEE Transactions on neural networks, 7(1), 229-232.

Cigizoglu, H. K. (2003). Estimation, forecasting and extrapolation of river flows by artificial neural networks. Hydrological Sciences Journal, 48(3), 349-361.

Cigizoglu, H. K. (2004). Estimation and forecasting of daily suspended sediment data by multi-layer perceptrons. Advances in Water Resources, 27(2), 185-195.

Cigizoglu, H. K. and Alp, M. (2003). Suspended sediment forecasting by artificial neural networks using hydro meteorological data. Paper presented at the World Water & Environmental Resources Congress 2003.

Cigizoglu, H. K. and Alp, M. (2006). Generalized regression neural network in modelling river sediment yield. Advances in Engineering Software, 37(2), 63-68.

Cigizoglu, H. K. and Kisi, Ö. (2006). Methods to improve the neural network performance in suspended sediment estimation. Journal of Hydrology, 317(3-4), 221-238.

Cimen, M. (2008). Estimation of daily suspended sediments using support vector machines. Hydrological Sciences Journal, 53(3), 656-666.

Clerc, M. and Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation, 6(1), 58-73.

Cobaner, M., Unal, B. and Kisi, Ö. (2009). Suspended sediment concentration estimation by an adaptive neuro-fuzzy and neural network approaches using hydro-meteorological data. Journal of Hydrology, 367(1), 52-61.

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

Coulibaly, P., Anctil, F. and Bobee, B. (2000). Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. Journal of Hydrology, 230(3), 244-257.

Crawford, C. G. (1991). Estimation of suspended-sediment rating curves and mean suspended-sediment loads. Journal of Hydrology, 129(1-4), 331-348.

Cruz-Vega, I., Garcia, C. A. R., Gil, P. G., Cortes, J. M. R., Magdaleno, J. D. R. and IEEE. (2016). Genetic algorithms based on a granular surrogate model and fuzzy aptitude functions. 2016 IEEE Congress on Evolutionary Computation (CEC), 2122-2128.

Dai, L. Q., Zhang, P. P., Wang, Y., Jiang, D. G., Dai, H. C., Mao, J. Q. and Wang, M. M. (2017). Multi-objective optimization of cascade reservoirs using NSGA-II: A case study of the Three Gorges-Gezhouba cascade reservoirs in the middle Yangtze River, China. Human and Ecological Risk Assessment, 23(4), 814-835.

Daliakopoulos, I. N. and Tsanis, I. K. (2016). Comparison of an artificial neural network and a conceptual rainfall-runoff model in the simulation of ephemeral streamflow. Hydrological Sciences Journal-Journal des Sciences Hydrologiques, 61(15), 2763-2774.

Danh, N. T., Phien, H. N. and Das Gupta, A. (1999). Neural network models for river flow forecasting. Water Sa, 25(1), 33-39.

Dawson, C., Harpham, C., Wilby, R. and Chen, Y. (2002). Evaluation of artificial neural network techniques for flow forecasting in the River Yangtze, China. Hydrology and Earth System Sciences Discussions, 6(4), 619-626.

De Vos, N. J. and Rientjes, T. H. M. (2005). Constraints of artificial neural networks for rainfall-runoff modelling: trade-offs in hydrological state representation and model evaluation. Hydrology and Earth System Sciences Discussions Discussions, 2(1), 365-415.

De Vos, N. J. and Rientjes, T. H. M. (2008). Multiobjective training of artificial neural networks for rainfall-runoff modeling. Water Resources Research, 44(8).

Dehghani, M., Saghafian, B., Nasiri Saleh, F., Farokhnia, A. and Noori, R. (2014). Uncertainty analysis of streamflow drought forecast using artificial neural networks and Monte-Carlo simulation. International Journal of Climatology, 34(4), 1169-1180.

Demirci, M. and Baltaci, A. (2013). Prediction of suspended sediment in river using fuzzy logic and multilinear regression approaches. Neural Computing and Applications, 23(1), 145-151.

Demirel, M. C., Booij, M. J. and Kahya, E. (2012). Validation of an ANN flow

prediction model using a multistation cluster analysis. Journal of Hydrologic Engineering, 17(2), 262-271.

Deng, Z., Bengtsson, L. and Singh, V. P. (2006). Parameter estimation for fractional dispersion model for rivers. Environmental Fluid Mechanics, 6(5), 451-475.

Dennis, J. E. and Schnabel, R. B. (1983). Numerical methods for unconstrained optimization and nonlinear equations: SIAM.

Dibike, Y. B. and Solomatine, D. P. (2001). River flow forecasting using artificial neural networks. Physics and Chemistry of the Earth, Part B: Hydrology, Oceans and Atmosphere, 26(1), 1-7.

Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. Journal of Business & Economic Statistics, 13(3), 253-263.

Dogan, E. (2009). Prediction of sediment concentration using artificial neural networks. Teknik Dergi, 20(1), 4567-4582.

Dogulu, N., Solomatine, D. and Shrestha, D. L. (2014). Applying clustering approach in predictive uncertainty estimation: a case study with the UNEEC method. Paper presented at the Geophysical Research Abstracts.

Doomen, A. M. C., Wijma, E., Zwolsman, J. J. G. and Middelkoop, H. (2008). Predicting suspended sediment concentrations in the Meuse River using a supply-based rating curve. Hydrological Processes, 22(12), 1846-1856.

Dorigo, M., Maniezzo, V. and Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 26(1), 29-41.

Ebrahimi, E., Monjezi, M., Khalesi, M. R. and Armaghani, D. J. (2016). Prediction and optimization of back-break and rock fragmentation using an artificial neural network and a bee colony algorithm. Bulletin of Engineering Geology and the Environment, 75(1), 27-36.

Ebrahimi, H., Jabbari, E. and Ghasemi, M. (2013). Application of honey-bees mating optimization algorithm on estimation of suspended sediment concentration. World Apply Sciences Journal, 22(11), 1630-1638.

Ercan, M. B. and Goodall, J. L. (2016). Design and implementation of a general software library for using NSGA-II with SWAT for multi-objective model calibration. Environmental Modelling & Software, 84, 112-120.

Ezzeldin, R., Djebedjian, B. and Saafan, T. (2014). Integer discrete particle swarm optimization of water distribution networks. Journal of Pipeline Systems Engineering and Practice, 5(1).

Faber, N. M. and Rajko, R. (2007). How to avoid over-fitting in multivariate calibration-The conventional validation approach and an alternative. Analytica Chimica Acta, 595(1), 98-106.

Ferket, B. V. A., Samain, B. and Pauwels, V. R. N. (2010). Internal validation of

conceptual rainfall-runoff models using baseflow separation. Journal of Hydrology, 381(1-2), 158-173.

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., Maier, H. R. and Dandy, G. C. (2009). Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach. Journal of Hydrology, 367(3-4), 165-176.

Firat, M. (2008). Comparison of artificial intelligence techniques for river flow forecasting. Hydrology and Earth System Sciences, 12(1), 123-139.

Firat, M. and Güngör, M. (2007). River flow estimation using adaptive neuro fuzzy inference system. Mathematics and Computers in Simulation, 75(3), 87-96.

Firat, M. and Güngör, M. (2010). Monthly total sediment forecasting using adaptive neuro fuzzy inference system. Stochastic Environmental Research and Risk Assessment, 24(2), 259-270.

Franchini, M. and Pacciani, M. (1991). Comparative analysis of several conceptual rainfall-runoff models. Journal of Hydrology, 122(1), 161-219.

Giustolisi, O. and Laucelli, D. (2005). Improving generalization of artificial neural networks in rainfall–runoff modelling. Hydrological Sciences Journal, 50(3), 439-457.

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

Glysson, G. D. (1987). Sediment-transport curves: US Geological Survey.

Golasowski, M., Litschmannová, M., Kuchar, S., Podhorányi, M. and Martinovic, J. (2015). Uncertainty modelling in Rainfall-Runoff simulations based on parallel Monte Carlo method. Neural Network World, 25(3), 267.

Golding, P., Kapadia, S., Naylor, S., Schulz, J., Maier, H. R., Lall, U. and van der Velde, M. (2017). Framework for minimising the impact of regional shocks on global food security using multi-objective ant colony optimisation. Environmental Modelling & Software, 95, 303-319.

Gopakumar, R., Takara, K. and James, E. J. (2007). Hydrologic data exploration and river flow forecasting of a humid tropical river basin using artificial neural networks. Water Resources Management, 21(11), 1915-1940.

Govindaraju, R. S. (2000). Artificial neural networks in hydrology. I: Preliminary concepts. Journal of Hydrologic Engineering, 5(2), 115-123.

Goyal, M. K. (2014). Modeling of sediment yield prediction using M5 model tree algorithm and wavelet regression. Water Resources Management, 28(7), 1991-2003.

Gozde, H. and Taplamacioglu, M. C. (2011). Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system. Journal of

the Franklin Institute-Engineering and Applied Mathematics, 348(8), 1927-1946.

Guldal, V. and Muftuoglu, R. F. (2001). 2D unit sediment graph theory. Journal of Hydrologic Engineering, 6(2), 132-140.

Guo, J., Zhou, J. Z., Song, L. X., Zou, Q. and Zeng, X. F. (2013). Uncertainty assessment and optimization of hydrological model with the Shuffled Complex Evolution Metropolis algorithm: an application to artificial neural network rainfall-runoff model. Stochastic Environmental Research and Risk Assessment, 27(4), 985-1004.

Guo, W. X. and Wang, H. X. (2010). PSO optimizing neural network for the Yangtze river sediment entering estuary prediction. Paper presented at the Natural Computation (ICNC), 2010 Sixth International Conference on.

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

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

Haddadchi, A. and Dehghani, A. A. (2012). Assessment of bed-load predictors based on sampling in a gravel bed river. Journal of Hydrodynamics, Ser. B, 24(1), 145-151.

Hagan, M. T. and Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. IEEE Transactions on neural networks, 5(6), 989-993.

Harrington, S. T. and Harrington, J. R. (2013). An assessment of the suspended sediment rating curve approach for load estimation on the Rivers Bandon and Owenabue, Ireland. Geomorphology, 185, 27-38.

Harvey, D., Leybourne, S. and Newbold, P. (1997). Testing the equality of prediction mean squared errors. International Journal of Forecasting, 13(2), 281-291.

Hassan, M., Shamim, M. A., Sikandar, A., Mehmood, I., Ahmed, I., Ashiq, S. Z. and Khitab, A. (2015). Development of sediment load estimation models by using artificial neural networking techniques. Environmental Monitoring and Assessment, 187(11).

He, J. X., Valeo, C., Chu, A. and Neumann, N. F. (2011). Prediction of event-based stormwater runoff quantity and quality by ANNs developed using PMI-based input selection. Journal of Hydrology, 400(1-2), 10-23.

He, M. (1993). Theory, application and related problems of double parallel feedforward neural networks. Xidian University, Xi'an.

He, M. Y. (1994). Double parallel feedforward neural networks with application to simulation study of flight fault inspection. Acta. Aeronautica. ET. Aeronautica. Sinica, 15, 877-881.

He, M. Y. and Huang, R. (2005). Feature selection for hyperspectral data classification using double parallel feedforward neural networks. Fuzzy Systems and Knowledge Discovery (pp. 58-66): Springer.

He, Q. and Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering Applications of Artificial Intelligence, 20(1), 89-99.

He, X. Z., Zhang, X. P., Zhang, S. J., Liu, J. D. and Li, C. S. (2005). Prediction of phase equilibrium properties for complicated macromolecular systems by HGALM neural networks. Fluid Phase Equilibria, 238(1), 52-57.

He, Y. J. and Ma, Z. F. (2014). Optimal design of linear sensor networks for process plants: A multi-objective ant colony optimization approach. Chemometrics and Intelligent Laboratory Systems, 135, 37-47.

He, Z. B., Wen, X. H., Liu, H. and Du, J. (2014a). A comparative study of artificial neural network, adaptive neuro fuzzy inference system and support vector machine for forecasting river flow in the semiarid mountain region. Journal of Hydrology, 509, 379-386.

He, Z. B., Zhao, W. Z. and Liu, H. (2014b). Comparing the performance of empirical black-box models for river flow forecasting in the Heihe River Basin, Northwestern China. Hydrological Processes, 28(1), 1-7.

Herbst, M., Casper, M. C., Grundmann, J. and Buchholz, O. (2009). Comparative analysis of model behaviour for flood prediction purposes using Self-Organizing Maps. Natural Hazards and Earth System Sciences, 9(2), 373-392.

Heydari, M., Othman, F. and Noori, M. (2016). Optimal operation of multiple and multipurpose reservoirs systems using non-dominated sorting genetic algorithm (NSGA-II). Fresenius Environmental Bulletin, 25(8), 2935-2946.

Himanshu, S. K., Pandey, A. and Yadav, B. (2017). Ensemble wavelet-support vector machine approach for prediction of suspended sediment load using hydrometeorological data. Journal of Hydrologic Engineering, 22(7).

Holland, J. H. (1975). Adaptation in natural and artificial systems. Ann Arbor, MI: University of Michigan Press

Horowitz, A. J. (2003). An evaluation of sediment rating curves for estimating suspended sediment concentrations for subsequent flux calculations. Hydrological Processes, 17(17), 3387-3409.

Hsu, K. L., Gupta, H. V. and Sorooshian, S. (1995). Arificial neural network modeling of the rainfall-runoff process. Water Resources Research, 31(10), 2517-2530.

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

Hu, T. S., Lam, K. C. and Ng, S. T. (2001). River flow time series prediction with a range-dependent neural network. Hydrological Sciences Journal 46(5), 729-745.

Huang, H., Qin, H., Hao, Z. F. and Lim, A. (2012). Example-based learning particle swarm optimization for continuous optimization. Information Sciences, 182(1), 125-138.

Humphrey, G. B., Gibbs, M. S., Dandy, G. C. and Maier, H. R. (2016). A hybrid approach to monthly streamflow forecasting: Integrating hydrological model outputs into a Bayesian artificial neural network. Journal of Hydrology, 540, 623-640.

Imrie, C. E., Durucan, S. and Korre, A. (2000). River flow prediction using artificial neural networks: generalisation beyond the calibration range. Journal of Hydrology, 233(1), 138-153.

Iniesta, A. A., Alcaraz, J. L. G. and Borbon, M. I. R. (2013). Optimization of injection molding process parameters by a hybrid of artificial neural network and artificial bee colony algorithm. Revista Facultad De Ingenieria-Universidad De Antioquia(67), 43-51.

Irani, R. and Nasimi, R. (2011). Application of artificial bee colony-based neural network in bottom hole pressure prediction in underbalanced drilling. Journal of Petroleum Science and Engineering, 78(1), 6-12.

Iritz, L. (1992). Rainfall input in an adaptive river flow forecast model. Hydrological Sciences Journal-Journal des Sciences Hydrologiques, 37(6), 607-620.

Ismail, S., Shabri, A. and Samsudin, R. (2012). A hybrid model of self organizing maps and least square support vector machine for river flow forecasting. Hydrology and Earth System Sciences, 16(11), 4417-4433.

Jacquin, A. P. and Shamseldin, A. Y. (2007). Development of a possibilistic method for the evaluation of predictive uncertainty in rainfall-runoff modeling. Water Resources Research, 43(4).

Jadon, S. S., Tiwari, R., Sharma, H. and Bansal, J. C. (2017). Hybrid artificial bee colony algorithm with differential evolution. Applied Soft Computing, 58, 11-24.

Jafrasteh, B. and Fathianpour, N. (2017). A hybrid simultaneous perturbation artificial bee colony and back propagation algorithm for training a local linear radial basis neural network on ore grade estimation. Neurocomputing, 235, 217-227.

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 Resources Research, 40(4).

Jain, A. and Srinivasulu, S. (2006). Integrated approach to model decomposed flow hydrograph using artificial neural network and conceptual techniques. Journal of Hydrology, 317(3), 291-306.

Jain, A., Sudheer, K. and Srinivasulu, S. (2004). Identification of physical processes inherent in artificial neural network rainfall runoff models. Hydrological Processes, 18(3), 571-581.

Jain, S. K. (2001). Development of integrated sediment rating curves using ANNs. Journal of Hydraulic Engineering, 127(1), 30-37.

Jain, S. K., Das, A. and Srivastava, D. K. (1999). Application of ANN for reservoir inflow prediction and operation. Journal of Water Resources Planning and Management-ASCE, 125(5), 263-271.

Jakubcova, M., Maca, P. and Pech, P. (2015). Parameter estimation in rainfall-runoff modelling using distributed versions of particle swarm optimization algorithm. Mathematical Problems in Engineering.

Jalali, M. R., Afshar, A. and Marino, M. A. (2006). Reservoir operation by ant colony optimization algorithms. Iranian Journal of Science and Technology, Transaction B: Engineering, 30(B1), 107-117.

Jansson, M. B. (1997). Comparison of sediment rating curves developed on load and on concentration. Nordic Hydrology, 28(3), 189-200.

Jena, P. P., Chatterjee, C., Pradhan, G. and Mishra, A. (2014). Are recent frequent high floods in Mahanadi basin in eastern India due to increase in extreme rainfalls? Journal of Hydrology, 517, 847-862.

Jha, S. K. and Bombardelli, F. A. (2011). Theoretical/numerical model for the transport of non-uniform suspended sediment in open channels. Advances in Water Resources, 34(5), 577-591.

Jia, G. L., Li, D. H., Yao, L. L., Zhao, P. C. and IEEE. (2016). An improved artificial bee colony-BP neural network algorithm in the short-term wind speed prediction. Proceedings of the 2016 12th World Congress on Intelligent Control and Automation (Wcica), 2252-2255.

Jiang, S. C., Chin, K. S., Wang, L., Qu, G. and Tsui, K. L. (2017). Modified genetic algorithm-based feature selection combined with pre-trained deep neural network for demand forecasting in outpatient department. Expert Systems with Applications, 82, 216-230.

Johnson, D., Smith, M., Koren, V. and Finnerty, B. (1999). Comparing mean areal precipitation estimates from NEXRAD and rain gauge networks. Journal of Hydrologic Engineering, 4(2), 117-124.

Joshi, R., Kumar, K. and Adhikari, V. P. S. (2016). Modelling suspended sediment concentration using artificial neural networks for Gangotri glacier. Hydrological Processes, 30(9), 1354-1366.

Jung, J., Seo, H., Yoo, I., Kim, H. and Kwon, S. (2015). Analysis of well testing data using ant colony optimization. Geosystem Engineering, 18(5), 266-271.

Kachroo, R. K. (1992). River flow forecasting. Part 1. A discussion of the principles. Journal of Hydrology, 133(1-2), 1-15.

Kalinli, A., Sagiroglu, S. and Sarikoc, F. (2010). Parallel ant colony optimization algorithm based neural method for determining resonant frequencies of various microstrip antennas. Electromagnetics, 30(5), 463-481.

Kan, G. Y., Yao, C., Li, Q. L., Li, Z. J., Yu, Z. B., Liu, Z. Y., . . . Liang, K. (2015). Improving event-based rainfall-runoff simulation using an ensemble artificial neural network based hybrid data-driven model. Stochastic environmental research and risk assessment, 29(5), 1345-1370.

Kang, K. W., Kim, J. H., Park, C. Y. and Ham, K. J. (1993). Evaluation of hydrologic

forecasting system based on neural network model. Paper presented at the The Congress-International Association for Hydraulic Research.

Kar, A. K., Lohani, A. K., Goel, N. K. and Roy, G. P. (2010). Development of flood forecasting system using statistical and ANN techniques in the downstream catchment of Mahanadi Basin, India. Journal of Water Resource and Protection, 2(10), 880.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical Report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 200.

Karaboga, D. and Akay, B. (2009). A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation, 214(1), 108-132.

Karaboga, D., Akay, B. and Ozturk, C. (2007). Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks Modeling Decisions for Artificial Intelligence (pp. 318-329): Springer Berlin Heidelberg.

Karaboga, D. and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization, 39(3), 459-471.

Karunanithi, N., Grenney, W. J., Whitley, D. and Bovee, K. (1994). Neural networks for river flow prediction. Journal of Computing in Civil Engineering, 8(2), 201-220.

Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. Paper presented at the Proceedings of ICNN'95 - International Conference on Neural Networks, New York.

Kerh, T. and Lee, C. (2006). Neural networks forecasting of flood discharge at an unmeasured station using river upstream information. Advances in Engineering Software, 37(8), 533-543.

Kesharaju, M. and Nagarajah, R. (2017). Particle swarm optimization approach to defect detection in armour ceramics. Ultrasonics, 75, 124-131.

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

Khan, A., Yang, J. and Wu, W. (2014). Double parallel feedforward neural network based on extreme learning machine with L 1/2 regularizer. Neuro Computing, 128, 113-118.

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

Khosravi, A., Mazloumi, E., Nahavandi, S., Creighton, D. and Van Lint, J. (2011a). Prediction intervals to account for uncertainties in travel time prediction. IEEE Transactions on Intelligent Transportation Systems, 12(2), 537-547.

Khosravi, A., Nahavandi, S. and Creighton, D. (2010). Construction of optimal prediction intervals for load forecasting problems. IEEE Transactions on Power Systems, 25(3), 1496-1503.

Khosravi, A., Nahavandi, S. and Creighton, D. (2011b). Prediction interval construction and optimization for adaptive neurofuzzy inference systems. IEEE Transactions on Fuzzy Systems, 19(5), 983-988.

Khosravi, A., Nahavandi, S., Creighton, D. and Atiya, A. F. (2011c). Lower upper bound estimation method for construction of neural network-based prediction intervals. Neural Networks, IEEE Transactions on, 22(3), 337-346.

Kim, S. and Kim, H. S. (2008). Uncertainty reduction of the flood stage forecasting using neural networks model. Journal of the American Water Resources Association, 44(1), 148-165.

Kingston, G. B., Lambert, M. F. and Maier, H. R. (2005). Bayesian training of artificial neural networks used for water resources modeling. Water Resources Research, 41(12).

Kisi, Ö. (2004a). Daily suspended sediment modelling using a fuzzy differential evolution approach. Hydrological Sciences Journal-Journal des Sciences Hydrologiques, 49(1), 183-197.

Kisi, Ö. (2004b). Multi-layer perceptrons with Levenberg-Marquardt training algorithm for suspended sediment concentration prediction and estimation. Hydrological Sciences Journal, 49(6), 1025-1040.

Kisi, Ö. (2004c). River flow modeling using artificial neural networks. Journal of Hydrologic Engineering, 9(1), 60-63.

Kisi, Ö. (2005a). Daily river flow forecasting using artificial neural networks and auto-regressive models. Turkish Journal of Engineering and Environmental Sciences, 29(1), 9-20.

Kisi, Ö. (2005b). Suspended sediment estimation using neuro-fuzzy and neural network approaches/Estimation des matières en suspension par des approches neurofloues et à base de réseau de neurones. Hydrological Sciences Journal, 50(4).

Kisi, Ö. (2008a). Constructing neural network sediment estimation models using a data-driven algorithm. Mathematics and Computers in Simulation, 79(1), 94-103.

Kisi, Ö. (2008b). River flow forecasting and estimation using different artificial neural network techniques. Hydrology Research, 39(1), 27-40.

Kisi, Ö. (2009). Evolutionary fuzzy models for river suspended sediment concentration estimation. Journal of Hydrology, 372(1-4), 68-79.

Kisi, Ö. (2010a). Daily suspended sediment estimation using neuro-wavelet models. International Journal of Earth Sciences, 99(6), 1471-1482.

Kisi, Ö. (2010b). River suspended sediment concentration modeling using a neural differential evolution approach. Journal of Hydrology, 389(1-2), 227-235.

Kisi, Ö. (2012). Modeling discharge-suspended sediment relationship using least square support vector machine. Journal of Hydrology, 456, 110-120.

Kisi, Ö. and Cigizoglu, H. K. (2007). Comparison of different ANN techniques in river

flow prediction. Civil Engineering and Environmental Systems, 24(3), 211-231.

Kisi, Ö. and Cimen, M. (2011). A wavelet-support vector machine conjunction model for monthly streamflow forecasting. Journal of Hydrology, 399(1), 132-140.

Kisi, Ö., Dailr, A. H., Cimen, M. and Shiri, J. (2012a). Suspended sediment modeling using genetic programming and soft computing techniques. Journal of Hydrology, 450, 48-58.

Kisi, Ö. and Fedakar, H. İ. (2014). Modeling of suspended sediment concentration carried in natural streams using fuzzy genetic approach. Computational Intelligence Techniques in Earth and Environmental Sciences (pp. 175-196): Springer.

Kisi, Ö. and Guven, A. (2010). A machine code-based genetic programming for suspended sediment concentration estimation. Advances in Engineering Software, 41(7-8), 939-945.

Kisi, Ö., Karahan, M. E. and Şen, Z. (2006). River suspended sediment modelling using a fuzzy logic approach. Hydrological Processes, 20(20), 4351-4362.

Kisi, Ö., Nia, A. M., Gosheh, M. G., Tajabadi, M. R. J. and Ahmadi, A. (2012b). Intermittent streamflow forecasting by using several data driven techniques. Water Resources Management, 26(2), 457-474.

Kisi, Ö., Ozkan, C. and Akay, B. (2012c). Modeling discharge-sediment relationship using neural networks with artificial bee colony algorithm. Journal of Hydrology, 428, 94-103.

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

Kisi, Ö., Yuksel, I. and Dogan, E. (2008). Modelling daily suspended sediment of rivers in Turkey using several data-driven techniques. Hydrological Sciences Journal-Journal des Sciences Hydrologiques, 53(6), 1270-1285.

Konar, M. and Bagis, A. (2016). Performance comparison of particle swarm optimization, differential evolution and artificial bee colony algorithms for fuzzy modelling of nonlinear systems. Elektronika Ir Elektrotechnika, 22(5), 8-13.

Korouzhdeh, T., Eskandari-Naddaf, H. and Gharouni-Nik, M. (2017). An improved ant colony model for cost optimization of composite beams. Applied Artificial Intelligence, 31(1), 44-63.

Kostic, S., Stojkovic, M. and Prohaska, S. (2016). Hydrological flow rate estimation using artificial neural networks: Model development and potential applications. Applied Mathematics and Computation, 291, 373-385.

Kothyari, U. C., Aravamuthan, V. and Singh, V. P. (1993). Monthly runoff generation using the linear perturbation model. Journal of Hydrology, 144(1-4), 371-379.

Kothyari, U. C., Tiwari, A. K. and Singh, R. (1997). Estimation of temporal variation of sediment yield from small catchments through the kinematic method. Journal of

Hydrology, 203(1), 39-57.

Kouassı, K. L., Kouame, K. I., Konan, K. S., Angulo, M. S. and Deme, M. (2013). Two-dimensional numerical simulation of the hydro-sedimentary phenomena in Lake Taabo, Côte d'ivoire. Water Resources Management, 27(12), 4379-4394.

Krzysztofowicz, R. (1999). Bayesian theory of probabilistic forecasting via deterministic hydrologic model. Water Resources Research, 35(9), 2739-2750.

Kuczera, G. and Parent, E. (1998). Monte Carlo assessment of parameter uncertainty in conceptual catchment models: the Metropolis algorithm. Journal of Hydrology, 211(1), 69-85.

Kumar, A. and Kumar, A. (2016). Adaptive management of multimodal biometrics fusion using ant colony optimization. Information Fusion, 32, 49-63.

Kumar, A. R. S., Ojha, C. S. P., Goyal, M. K., Singh, R. D. and Swamee, P. K. (2012). Modeling of suspended sediment concentration at Kasol in India using ANN, Fuzzy Logic, and Decision Tree Algorithms. Journal of Hydrologic Engineering, 17(3), 394-404.

Kumar, D., Pandey, A., Sharma, N. and Flugel, W. A. (2015a). Modeling suspended sediment using artificial neural networks and TRMM-3B42 Version 7 rainfall dataset. Journal of Hydrologic Engineering, 20(6).

Kumar, D., Pandey, A., Sharma, N. and Flugel, W. A. (2016). Daily suspended sediment simulation using machine learning approach. Catena, 138, 77-90.

Kumar, D. N. and Reddy, M. J. (2006). Ant colony optimization for multi-purpose reservoir operation. Water Resources Management, 20(6), 879-898.

Kumar, D. N. and Reddy, M. J. (2007). Multipurpose reservoir operation using particle swarm optimization. Journal of Water Resources Planning and Management-Asce, 133(3), 192-201.

Kumar, S., Tiwari, M. K., Chatterjee, C. and Mishra, A. (2015b). Reservoir inflow forecasting using ensemble models based on neural networks, wavelet analysis and bootstrap method. Water Resources Management, 29(13), 4863-4883.

Kundzewicz, Z. W. (2007). New uncertainty concepts in hydrology and water resources: Cambridge University Press.

Lafdani, E. K., Nia, A. M. and Ahmadi, A. (2013). Daily suspended sediment load prediction using artificial neural networks and support vector machines. Journal of Hydrology, 478, 50-62.

Lahiri, S. K. and Khalfe, N. M. (2015). Hybrid particle swarm optimization and ant colony optimization technique for the optimal design of shell and tube heat exchangers. Chemical Product and Process Modeling, 10(2), 81-96.

Lampinen, J. and Vehtari, A. (2001). Bayesian approach for neural networks-review and case studies. Neural Networks, 14(3), 257-274.

Latt, Z. Z. and Wittenberg, H. (2014). Improving flood forecasting in a developing country: a comparative study of stepwise multiple linear regression and artificial neural network. Water Resources Management, 28(8), 2109-2128.

Lazzus, J. A., Rivera, M., Salfate, I., Pulgar-Villarroel, G. and Rojas, P. (2016). Application of particle swarm plus ant colony optimization to calculate the interaction parameters on phase equilibria. Journal of Engineering Thermophysics, 25(2), 216-226.

Leahy, P., Kiely, G. and Corcoran, G. (2008). Structural optimisation and input selection of an artificial neural network for river level prediction. Journal of Hydrology, 355(1), 192-201.

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

Li, J. B. and Chung, Y. K. (2005). A novel back-propagation neural network training algorithm designed by an ant colony optimization. Paper presented at the Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES.

Li, L. Y., Chen, Y., Yu, X., Liu, R. and Huang, C. (2015). Sub-pixel flood inundation mapping from multispectral remotely sensed images based on discrete particle swarm optimization. Isprs Journal of Photogrammetry and Remote Sensing, 101, 10-21.

Li, M. and Chen, S. Y. (2010). Fuzzy variable classified method and its application in basin floods Fuzzy Information and Engineering 2010 (pp. 627-636): Springer.

Li, X. T., Liu, H. W. and Yin, M. H. (2013a). Differential evolution for prediction of longitudinal dispersion coefficients in natural streams. Water Resources Management, 27(15), 5245-5260.

Li, Y. C., Wang, Y. L. and Li, B. (2013b). A hybrid artificial bee colony assisted differential evolution algorithm for optimal reactive power flow. International Journal of Electrical Power & Energy Systems, 52, 25-33.

Li, Y. T. 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, Z. X., Janardhanan, M. N., Tang, Q. H. and Nielsen, P. (2016). Co-evolutionary particle swarm optimization algorithm for two-sided robotic assembly line balancing problem. Advances in Mechanical Engineering, 8(9).

Lin, B. and Namin, M. M. (2005). Modelling suspended sediment transport using an integrated numerical and ANNs model. Journal of Hydraulic Research, 43(3), 302-310.

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

Linares-Rodriguez, A., Lara-Fanego, V., Pozo-Vazquez, D. and Tovar-Pescador, J. (2015). One-day-ahead streamflow forecasting using artificial neural networks and a meteorological mesoscale model. Journal of Hydrologic Engineering, 20(9).

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(3), 431-448.

Liu, F. F., Li, Y. P., Huang, G. H., Cui, L. and Liu, J. (2016). Assessing uncertainty in hydrological processes using a fuzzy vertex simulation method. Journal of Hydrologic Engineering, 21(4), 05016002.

Liu, J., Ma, D., Ma, T. B. and Zhang, W. (2017a). Ecosystem particle swarm optimization. Soft Computing, 21(7), 1667-1691.

Liu, M. T., Zhang, C. C., Liu, X. M., Yuan, S. and Jiang, E. H. (2015). The study of data fusion for high suspended sediment concentration measuring using the IGA-RBF method. Journal of Intelligent & Fuzzy Systems, 28(2), 605-614.

Liu, Q. J., Shi, Z. H., Fang, N. F., Zhu, H. D. and Ai, L. (2013a). Modeling the daily suspended sediment concentration in a hyperconcentrated river on the Loess Plateau, China, using the Wavelet-ANN approach. Geomorphology, 186, 181-190.

Liu, S., Xu, J. W., Zhao, J. F. and Li, Y. (2013b). A novel integrated rainfall-runoff model based on TOPMODEL and artificial neural network. Paper presented at the Applied Mechanics and Materials.

Liu, Y. (2009). Automatic calibration of a rainfall-runoff model using a fast and elitist multi-objective particle swarm algorithm. Expert Systems with Applications, 36(5), 9533-9538.

Liu, Y. and Pender, G. (2013). Automatic calibration of a rapid flood spreading model using multiobjective optimisations. Soft Computing, 17(4), 713-724.

Liu, Y., Wang, X. D., Du, F. M., Yao, M., Gao, Y. L., Wang, F. W. and Wang, J. Y. (2017b). Computer vision detection of mold breakout in slab continuous casting using an optimized neural network. International Journal of Advanced Manufacturing Technology, 88(1-4), 557-564.

Lohani, A., Goel, N. and Bhatia, K. S. (2007). Deriving stage-discharge-sediment concentration relationships using fuzzy logic. Hydrological Sciences Journal, 52(4), 793-807.

Lohani, A. K., Goel, N. and Bhatia, K. (2014). Improving real time flood forecasting using fuzzy inference system. Journal of Hydrology, 509, 25-41.

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

Londhe, S. and Gavraskar, S. S. (2015). Forecasting one day ahead stream flow using support vector regression. In G. S. Dwarakish (Ed.), International Conference on Water Resources, Coastal and Ocean Engineering (Vol. 4, pp. 900-907).

Lopez, P. R., Galan, S. G., Reyes, N. R. and Jurado, F. (2008). A method for particle swarm optimization and its application in location of biomass power plants. International Journal of Green Energy, 5(3), 199-211.

Lu, H. S., Hou, T., Horton, R., Zhu, Y. H., Chen, X., Jia, Y. W., . . . Fu, X. L. (2013). The streamflow estimation using the Xinanjiang rainfall runoff model and dual state-parameter estimation method. Journal of Hydrology, 480, 102-114.

Lu, J. N., Hu, H. P. and Bai, Y. P. (2015). Generalized radial basis function neural network based on an improved dynamic particle swarm optimization and AdaBoost algorithm. Neurocomputing, 152, 305-315.

Lu, Z. B., Zhang, A. and Hou, X. Y. (2008). Pattern synthesis of cylindrical conformal array by the modified particle swarm optimization algorithm. Progress in Electromagnetics Research-Pier, 79, 415-426.

Mahmoudabadi, H., Izadi, M. and Menhaj, M. B. (2009). A hybrid method for grade estimation using genetic algorithm and neural networks. Computational Geosciences, 13(1), 91-101.

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.

Maier, H. R., Simpson, A. R., Zecchin, A. C., Foong, W. K., Phang, K. Y., Seah, H. Y. and Tan, C. L. (2003). Ant colony optimization for design of water distribution systems. Journal of Water Resources Planning and Management, 129(3), 200-209.

Malik, A., Kumar, A. and Piri, J. (2017). Daily suspended sediment concentration simulation using hydrological data of Pranhita River Basin, India. Computers and Electronics in Agriculture, 138, 20-28.

Mantovan, P. and Todini, E. (2006). Hydrological forecasting uncertainty assessment: Incoherence of the GLUE methodology. Journal of Hydrology, 330(1), 368-381.

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

Masoumeh, R. and Mehdi, F. (2012). Estimating suspended sediment concentration by a neural differential evolution (NDE) and comparison to ANFIS and three ANN models. Disaster Advances, 5(4), 346-359.

Masuda, K. and Kurihara, K. (2012). A constrained global optimization method based on multi-objective particle swarm optimization. Electronics and Communications in Japan, 95(1), 43-54.

Mavrovouniotis, M. and Yang, S. X. (2015). Training neural networks with ant colony optimization algorithms for pattern classification. Soft Computing, 19(6), 1511-1522.

May, R. J., Dandy, G. C., Maier, H. R. and Nixon, J. B. (2008). Application of partial mutual information variable selection to ANN forecasting of water quality in water distribution systems. Environmental Modelling & Software, 23(10-11), 1289-1299.

McBean, E. A. and Al-Nassri, S. (1988). Uncertainty in suspended sediment transport curves. Journal of Hydraulic Engineering, 114(1), 63-74.

McInerney, D., Thyer, M., Kavetski, D., Lerat, J. and Kuczera, G. (2017). Improving probabilistic prediction of daily streamflow by identifying Pareto optimal approaches for modeling heteroscedastic residual errors. Water Resources Research, 53(3), 2199-2239.

McIntyre, N. and Al-Qurashi, A. (2009). Performance of ten rainfall-runoff models applied to an arid catchment in Oman. Environmental Modelling & Software, 24(6), 726-738.

Melesse, A. M., Ahmad, S., McClain, M. E., Wang, X. and Lim, Y. H. (2011). Suspended sediment load prediction of river systems: An artificial neural network approach. Agricultural Water Management, 98(5), 855-866.

Mendes, R., Kennedy, J. and Neves, J. (2004). The fully informed particle swarm: Simpler, maybe better. IEEE Transactions on Evolutionary Computation, 8(3), 204-210.

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

Mirbagheri, S. A., Nourani, V., Rajaee, T. and Alikhani, A. (2010). Neuro-fuzzy models employing wavelet analysis for suspended sediment concentration prediction in rivers. Hydrological Sciences Journal-Journal des Sciences Hydrologiques, 55(7), 1175-1189.

Mirzaee, H. (2009). Linear combination rule in genetic algorithm for optimization of finite impulse response neural network to predict natural chaotic time series. Chaos Solitons & Fractals, 41(5), 2681-2689.

Montalvo, I., Izquierdo, J., Perez-Garcia, R. and Herrera, M. (2010). Improved performance of PSO with self-adaptive parameters for computing the optimal design of Water Supply Systems. Engineering Applications of Artificial Intelligence, 23(5), 727-735.

Montalvo, I., Izquierdo, J., Perez, R. and Tung, M. M. (2008). Particle swarm optimization applied to the design of water supply systems. Computers & Mathematics with Applications, 56(3), 769-776.

Montanari, A. (2007). What do we mean by 'uncertainty'? The need for a consistent wording about uncertainty assessment in hydrology. Hydrological Processes, 21(6), 841-845.

Montanari, A. and Brath, A. (2004). A stochastic approach for assessing the uncertainty of rainfall-runoff simulations. Water Resources Research, 40(1).

Montanari, A. and Grossi, G. (2008). Estimating the uncertainty of hydrological forecasts: A statistical approach. Water Resources Research, 44(12).

Moore, R. J. (2007). The PDM rainfall-runoff model. Hydrology and Earth System Sciences Discussions, 11(1), 483-499.

Moore, R. J. and Bell, V. A. (2002). Incorporation of groundwater losses and well level data in rainfall-runoff models illustrated using the PDM. Hydrology and Earth System Sciences, 6(1), 25-38.

Mustafa, M. R., Rezaur, R. B., Saiedi, S. and Isa, M. H. (2012). River suspended sediment prediction using various multilayer perceptron neural network training algorithms-a case study in Malaysia. Water Resources Management, 26(7), 1879-1897.

Najafzadeh, M. and Zahiri, A. (2015). Neuro-fuzzy GMDH-based evolutionary algorithms to predict flow discharge in straight compound channels. Journal of Hydrologic Engineering, 20(12).

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.

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

Nayak, P. C., Venkatesh, B., Krishna, B. and Jain, S. K. (2013). Rainfall-runoff modeling using conceptual, data driven, and wavelet based computing approach. Journal of Hydrology, 493, 57-67.

Nazemi, A., Yao, X., Chan, A. H. and IEEE. (2006). Extracting a set of robust pareto-optimal parameters for hydrologic models using NSGA-II and SCEM. Paper presented at the 2006 IEEE Congress on Evolutionary Computation, Vols 1-6, New York.

Nguyen, D. C. H., Dandy, G. C., Maier, H. R. and Ascough, J. C. (2016). Improved ant colony optimization for optimal crop and irrigation Water allocation by incorporating domain knowledge. Journal of Water Resources Planning and Management, 142(9).

Nguyen, P. K. T., Chua, L. H. C. and Son, L. H. (2014). Flood forecasting in large rivers with data-driven models. Natural Hazards, 71(1), 767-784.

Ning, A. P., Zhang, X. Y. and IEEE. (2011). A speech recognition system based on fuzzy neural network trained by artificial bee colony algorithm. 2011 International Conference on Electronics, Communications and Control (ICECC), 2488-2491.

Noh, S. J., Rakovec, O., Weerts, A. H. and Tachikawa, Y. (2014). On noise specification in data assimilation schemes for improved flood forecasting using distributed hydrological models. Journal of Hydrology, 519, 2707-2721.

Noori, R., Farokhnia, A., Morid, S. and Riahi Madvar, H. (2009). Effect of input variables preprocessing in artificial neural network on monthly flow prediction by PCA and wavelet transformation. J. of Water and Wastewater, 1, 13-22.

Noori, R., Karbassi, A. R., Moghaddamnia, A., Han, D., Zokaei-Ashtiani, M. H., Farokhnia, A. and Gousheh, M. G. (2011). Assessment of input variables determination on the SVM model performance using PCA, Gamma test, and forward selection techniques for monthly stream flow prediction. Journal of Hydrology, 401(3-4), 177-189.

Nourani, V. and Andalib, G. (2015). Daily and monthly suspended sediment load predictions using wavelet based artificial intelligence approaches. Journal of Mountain Science, 12(1), 85-100.

Nourani, V. and Komasi, M. (2013). A geomorphology-based ANFIS model for multi-station modeling of rainfall–runoff process. Journal of Hydrology, 490, 41-55.

Nourani, V., Komasi, M. and Mano, A. (2009). A multivariate ANN-wavelet approach for rainfall–runoff modeling. Water Resources Management, 23(14), 2877-2894.

Nourani, V., Mogaddam, A. A. and Nadiri, A. O. (2008). An ANN-based model for spatiotemporal groundwater level forecasting. Hydrological Processes, 22(26), 5054-5066.

Odiyo, J. O., Phangisa, J. I. and Makungo, R. (2012). Rainfall-runoff modelling for estimating Latonyanda River flow contributions to Luvuvhu River downstream of Albasini Dam. Physics and Chemistry of the Earth, Parts A/B/C, 50, 5-13.

Okkan, U. (2012). Using wavelet transform to improve generalization capability of feed forward neural networks in monthly runoff prediction. Scientific Research and Essays, 7(17), 1690-1703.

Ouyang, Y. (2005). Evaluation of river water quality monitoring stations by principal component analysis. Water Research, 39(12), 2621-2635.

Ozkan, C., Kisi, Ö. and Akay, B. (2011). Neural networks with artificial bee colony algorithm for modeling daily reference evapotranspiration. Irrigation Science, 29(6), 431-441.

Ozyurt, N. N. and Sezer, E. A. (2012). Daily streamflow prediction by ANFIS modeling: Application to Lower Zamanti Karst Basin, Turkey. Journal of Intelligent & Fuzzy Systems, 23(6), 305-311.

Panagoulia, D., Tsekouras, G. J. and Kousiouris, G. (2017). A multi-stage methodology for selecting input variables in ANN forecasting of river flows. Global Nest Journal, 19(1), 49-57.

Pang, S. C., Ma, T. M. and Liu, T. (2015). An improved ant colony optimization with optimal search library for solving the traveling salesman problem. Journal of Computational and Theoretical Nanoscience, 12(7), 1440-1444.

Partal, T. (2008). River flow forecasting using different artificial neural network algorithms and wavelet transform. Canadian Journal of Civil Engineering, 36(1), 26-38.

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

Partovian, A., Nourani, V. and Alami, M. T. (2016). Hybrid denoising-jittering data processing approach to enhance sediment load prediction of muddy rivers. Journal of Mountain Science, 13(12), 2135-2146.

Patel, S. S. and Ramachandran, P. (2015). A comparison of machine learning techniques for modeling river flow time series: the case of Upper Cauvery River Basin. Water Resources Management, 29(2), 589-602.

Pawde, A. W., Mathur, Y. P. and Kumar, R. (2013). Optimal water scheduling in irrigation canal network using particle swarming optimization. Irrigation and Drainage,

62(2), 135-144.

Pektas, A. O. (2015). Determining the essential parameters of bed load and suspended sediment load. International Journal of Global Warming, 8(3), 335-359.

Peng, H. C., Long, F. H. and Ding, C. (2005). Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8), 1226-1238.

Peng, Y. (2012). An improved particle swarm optimization algorithm for optimal operation of cascade reservoirs. Information-an International Interdisciplinary Journal, 15(1), 33-40.

Peng, Y. and Liang, G. H. (2009). A fuzzy optimization neural network model using second order information. Paper presented at the Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on.

Pianosi, F., Lal Shrestha, D. and Solomatine, D. (2010). Uncertainty analysis of an inflow forecasting model: extension of the UNEEC machine learning-based method. Paper presented at the EGU General Assembly Conference Abstracts.

Piotrowski, A., Wallis, S. G., Napiorkowski, J. J. and Rowinski, P. M. (2007). Evaluation of 1-D tracer concentration profile in a small river by means of Multi-Layer Perceptron Neural Networks. Hydrology and Earth System Sciences 11, 1883-1896.

Piotrowski, A. P. and Napiorkowski, J. J. (2011). Optimizing neural networks for river flow forecasting- evolutionary computation methods versus the Levenberg-Marquardt approach. Journal of Hydrology, 407(1-4), 12-27.

Piotrowski, A. P., Rowinski, P. M. and Napiorkowski, J. J. (2012). Comparison of evolutionary computation techniques for noise injected neural network training to estimate longitudinal dispersion coefficients in rivers. Expert Systems with Applications, 39(1), 1354-1361.

Pramanik, N., Panda, R. and Singh, A. (2011). Daily river flow forecasting using wavelet ANN hybrid models. Journal of Hydroinformatics, 13(1), 49-63.

Pramanik, N. and Panda, R. K. (2009). Application of neural network and adaptive neuro-fuzzy inference systems for river flow prediction. Hydrological Sciences Journal, 54(2), 247-260.

Prudencio, R. B. C. and Ludermir, T. B. (2003). Neural network hybrid learning: Genetic Algorithms & Levenberg-Marquardt. In M. Schader, W. Gaul & M. Vichi (Eds.), Between Data Science and Applied Data Analysis (pp. 464-472). Berlin: Springer-Verlag Berlin.

Qin, H., Zhou, J. Z., Lu, Y. L., Li, Y. H. and Zhang, Y. C. (2010). Multi-objective cultured differential evolution for generating optimal trade-offs in reservoir flood control operation. Water Resources Management, 24(11), 2611-2632.

Qiu, J. H., Chen, R. B., Wang, W. C. and Wong, W. K. (2014). Using animal instincts to design efficient biomedical studies via particle swarm optimization. Swarm and Evolutionary Computation, 18, 1-10.

Qiu, L., Chen, S. Y. and Nie, X. T. (1998). A forecast model of fuzzy recognition neural networks and its application. Advances in Water Science, 9(3), 258-264.

Quan, H., Srinivasan, D. and Khosravi, A. (2014a). Particle swarm optimization for construction of neural network-based prediction intervals. Neurocomputing, 127, 172-180.

Quan, H., Srinivasan, D. and Khosravi, A. (2014b). Short-term load and wind power forecasting using neural network-based prediction intervals. IEEE Transactions on Neural Networks and Learning Systems, 25(2), 303-315.

Quan, H., Srinivasan, D. and Khosravi, A. (2014c). Uncertainty handling using neural network-based prediction intervals for electrical load forecasting. Energy, 73, 916-925.

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

Rajaee, T. (2010). Wavelet and neuro-fuzzy conjunction approach for suspended sediment prediction. Clean-Soil Air Water, 38(3), 275-288.

Rajaee, T. (2011). Wavelet and ANN combination model for prediction of daily suspended sediment load in rivers. Science of the Total Environment, 409(15), 2917-2928.

Rajaee, T., Mirbagheri, S. A., Nourani, V. and Alikhani, A. (2010). Prediction of daily suspended sediment load using wavelet and neuro-fuzzy combined model. International Journal of Environmental Science and Technology, 7(1), 93-110.

Rajaee, T., Mirbagheri, S. A., Zounemat-Kermani, M. and Nourani, V. (2009). Daily suspended sediment concentration simulation using ANN and neuro-fuzzy models. Science of the total environment, 407(17), 4916-4927.

Rajaee, T., Nourani, V., Zounemat-Kermani, M. and Kisi, Ö. (2011). River suspended sediment load prediction: application of ANN and wavelet conjunction model. Journal of Hydrologic Engineering, 16(8), 613-627.

Rakitianskaia, A. S. and Engelbrecht, A. P. (2012). Training feedforward neural networks with dynamic particle swarm optimisation. Swarm Intelligence, 6(3), 233-270.

Raman, H. and Sunilkumar, N. (1995). Multivariate modeling of water resources time series using artificial neural networks. Hydrological Sciences Journal-Journal des Sciences Hydrologiques, 40(2), 145-163.

Rana, M., Koprinska, I., Khosravi, A. and Agelidis, V. G. (2013). Prediction intervals for electricity load forecasting using neural networks. Paper presented at the Neural Networks (IJCNN), The 2013 International Joint Conference on.

Rana, S., Jasola, S. and Kumar, R. (2011). A review on particle swarm optimization algorithms and their applications to data clustering. Artificial Intelligence Review, 35(3), 211-222.

Rashidi, S., Vafakhah, M., Lafdani, E. K. and Javadi, M. R. (2016). Evaluating the

support vector machine for suspended sediment load forecasting based on gamma test. Arabian Journal of Geosciences, 9(11).

Rech, G. (2002). Forecasting with artificial network models: SSE/EFI Working Paper Series in Economics and Finance 491.

Reddy, M. J. and Kumar, D. N. (2007). Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation. Hydrological Processes, 21(21), 2897-2909.

Reddy, M. J. and Kumar, D. N. (2008). Evolving strategies for crop planning and operation of irrigation reservoir system using multi-objective differential evolution. Irrigation Science, 26(2), 177-190.

Remesan, R., Ahmadi, A., Shamim, M. A. and Han, D. W. (2010). Effect of data time interval on real-time flood forecasting. Journal of Hydroinformatics, 12(4), 396-407.

Reyhani, N., Hao, J., Ji, Y. and Lendasse, A. (2005). Mutual information and gamma test for input selection. Paper presented at the European Symposium Artifical Neural Networks Bruges 2005, Belgium.

Rezaeian-Zadeh, M., Tabari, H. and Abghari, H. (2013). Prediction of monthly discharge volume by different artificial neural network algorithms in semi-arid regions. Arabian Journal of Geosciences, 6(7), 2529-2537.

Rezaeianzadeh, M., Stein, A., Tabari, H., Abghari, H., Jalalkamali, N., Hosseinipour, E. Z. and Singh, V. P. (2013). Assessment of a conceptual hydrological model and artificial neural networks for daily outflows forecasting. International Journal of Environmental Science and Technology, 10(6), 1181-1192.

Rezaeianzadeh, M., Tabari, H., Yazdi, A. A., Isik, S. and Kalin, L. (2014). Flood flow forecasting using ANN, ANFIS and regression models. Neural Computing and Applications, 25(1), 25-37.

Rezaie-Balf, M., Zahmatkesh, Z. and Kim, S. (2017). Soft computing techniques for rainfall-runoff simulation: local non-parametric paradigm vs. model classification methods. Water Resources Management, 31(12), 3843-3865.

Riley, J. and Ciesielski, V. B. (1998). An evolutionary approach to training feed-forward and recurrent neural networks. New York: IEEE.

Rocca, P., Oliveri, G. and Massa, A. (2011). Differential evolution as applied to electromagnetics. Antennas and Propagation Magazine, IEEE, 53(1), 38-49.

Roushangar, K., Hosseinzadeh, S. and Shiri, J. (2016). Local vs. cross station simulation of suspended sediment load in successive hydrometric stations: heuristic modeling approach. Journal of Mountain Science, 13(10), 1773-1788.

Rout, M., Majhi, B., Mohapatra, U. M. and Mahapatra, R. (2012). An artificial bee colony algorithm based efficient prediction model for stock market indices. Paper presented at the Proceedings of the 2012 World Congress on Information and Communication Technologies, New York.

Rumelhart, D. E. and Mcclelland, J. L. (1986). Parallel distributed processing: Explorations in the microstructure of cognition: Foundations (Parallel distributed processing): MIT Press, August.

Sajikumar, N. and Thandaveswara, B. S. (1999). A non-linear rainfall-runoff model using an artificial neural network. Journal of Hydrology, 216(1-2), 32-55.

Samain, B. and Pauwels, V. R. N. (2013). Impact of potential and (scintillometer-based) actual evapotranspiration estimates on the performance of a lumped rainfall-runoff model. Hydrology and Earth System Sciences, 17(11), 4525-4540.

Savic, D. A., Walters, G. A. and Davidson, J. W. (1999). A genetic programming approach to rainfall-runoff modelling. Water Resources Management, 13(3), 219-231.

Schleiter, I. M., Borchardt, D., Wagner, R., Dapper, T., Schmidt, K. D., Schmidt, H. H. and Werner, H. (1999). Modelling water quality, bioindication and population dynamics in lotic ecosystems using neural networks. Ecological Modelling, 120(2-3), 271-286.

Sebt, M. H., Afshar, M. R. and Alipouri, Y. (2017). Hybridization of genetic algorithm and fully informed particle swarm for solving the multi-mode resource-constrained project scheduling problem. Engineering Optimization, 49(3), 513-530.

Sen, Z. (1998). Average areal precipitation by percentage weighted polygon method. Journal of Hydrologic Engineering, 3(1), 69-72.

Senthil Kumar, A., Ojha, C., Goyal, M. K., Singh, R. and Swamee, P. (2011). Modeling of suspended sediment concentration at Kasol in India using ANN, Fuzzy Logic, and Decision Tree algorithms. Journal of Hydrologic Engineering, 17(3), 394-404.

Setiono, R. and Hui, L. C. K. (1995). Use of a quasi-Newton method in a feedforward neural network construction algorithm. IEEE Transactions on neural networks, 6(1), 273-277.

Shamaei, E. and Kaedi, M. (2016). Suspended sediment concentration estimation by stacking the genetic programming and neuro-fuzzy predictions. Applied Soft Computing, 45, 187-196.

Shamseldin, A. Y. (2010). Artificial neural network model for river flow forecasting in a developing country. Journal of Hydroinformatics, 12(1), 22-35.

Sharma, S. and Tiwari, K. (2009). Bootstrap based artificial neural network (BANN) analysis for hierarchical prediction of monthly runoff in Upper Damodar Valley Catchment. Journal of Hydrology, 374(3), 209-222.

Shelokar, P. S., Siarry, P., Jayaraman, V. K. and Kulkarni, B. D. (2007). Particle swarm and ant colony algorithms hybridized for improved continuous optimization. Applied Mathematics and Computation, 188(1), 129-142.

Sheta, A. F. and Mahmoud, A. (2001). Forecasting using genetic programming. Paper presented at the Southeastern Symposium on System Theory.

Shiri, J. and Kisi, Ö. (2012). Estimation of daily suspended sediment load by using wavelet conjunction models. Journal of Hydrologic Engineering, 17(9), 986-1000.

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.

Si, W., Bao, W. M., Jiang, P., Zhao, L. P. and Qu, S. M. (2017). A semi-physical sediment yield model for estimation of suspended sediment in loess region. International Journal of Sediment Research, 32(1), 12-19.

Sigarchian, S. G., Orosz, M. S., Hemond, H. F. and Malmquist, A. (2016). Optimum design of a hybrid PV-CSP-LPG microgrid with Particle Swarm Optimization technique. Applied Thermal Engineering, 109, 1031-1036.

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

Singh, A., Imtiyaz, M., Isaac, R. K. and Denis, D. M. (2013). Comparison of artificial neural network models for sediment yield prediction at single gauging station of watershed in Eastern India. Journal of Hydrologic Engineering, 18(1), 115-120.

Singh, P. and Deo, M. (2007). Suitability of different neural networks in daily flow forecasting. Applied Soft Computing, 7(3), 968-978.

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

Sleigh, P., Gaskell, P., Berzins, M. and Wright, N. (1998). An unstructured finite-volume algorithm for predicting flow in rivers and estuaries. Computers & Fluids, 27(4), 479-508.

Socha, K. and Blum, C. (2007). An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training. Neural Computing and Applications, 16(3), 235-247.

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 Shrestha, D. L. (2009). A novel method to estimate model uncertainty using machine learning techniques. Water Resources Research, 45(12).

Song, X. H., Li, L., Zhang, X. Q., Huang, J. Q., Shi, X. C., Jin, S. and Bai, Y. M. (2014). Differential evolution algorithm for nonlinear inversion of high-frequency Rayleigh wave dispersion curves. Journal of Applied Geophysics, 109, 47-61.

Song, X. M., Kong, F. Z., Zhan, C. S. and Han, J. W. (2011). Hybrid optimization rainfall-runoff simulation based on Xinanjiang model and artificial neural network. Journal of Hydrologic Engineering, 17(9), 1033-1041.

Storn, R. and Price, K. (1995). Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. Paper presented at the International Computer Science Institute, Berkeley.

Subudhi, B. and Jena, D. (2008). Differential evolution and levenberg marquardt trained

neural network scheme for nonlinear system identification. Neural Processing Letters, 27(3), 285-296.

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

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(6), 1325-1330.

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

Sudheer, K. P., Nayak, P. C. and Ramasastri, K. S. (2003). Improving peak flow estimates in artificial neural network river flow models. Hydrological Processes, 17(3), 677-686.

Suribabu, C. R. and Neelakantan, T. R. (2006). Particle swarm optimization compared to other heuristic search techniques for pipe sizing. Journal of Environmental Informatics, 8(1), 1-9.

Szemis, J. M., Maier, H. R. and Dandy, G. C. (2012). A framework for using ant colony optimization to schedule environmental flow management alternatives for rivers, wetlands, and floodplains. Water Resources Research, 48.

Tabari, M. M. R. (2016). Prediction of river runoff using fuzzy theory and direct search optimization algorithm coupled model. Arabian Journal for Science and Engineering, 41(10), 4039-4051.

Taheri, K., Hasanipanah, M., Golzar, S. B. and Abd Majid, M. Z. (2017). A hybrid artificial bee colony algorithm-artificial neural network for forecasting the blast-produced ground vibration. Engineering with Computers, 33(3), 689-700.

Taormina, R. and Chau, K. W. (2015a). ANN-based interval forecasting of streamflow discharges using the LUBE method and MOFIPS. Engineering Applications of Artificial Intelligence, 45, 429-440.

Taormina, R. and Chau, K. W. (2015b). Data-driven input variable selection for rainfall-runoff modeling using binary-coded particle swarm optimization and Extreme Learning Machines. Journal of Hydrology, 529, 1617-1632.

Taormina, R. and Chau, K. W. (2015c). Neural network river forecasting with multi-objective fully informed particle swarm optimization. Journal of Hydroinformatics, 17(1), 99-113.

Tayfur, G., Karimi, Y. and Singh, V. P. (2013). Principle component analysis in conjuction with data driven methods for sediment load prediction. Water Resources Management, 27(7), 2541-2554.

Tayfur, G. and Moramarco, T. (2008). Predicting hourly-based flow discharge hydrographs from level data using genetic algorithms. Journal of Hydrology, 352(1-2), 77-93.

Tehzeeb-Ul-Hassan, H., Zafar, R., Mohsin, S. A. and Lateef, O. (2012). Reduction in power transmission loss using fully informed particle swarm optimization. International Journal of Electrical Power & Energy Systems, 43(1), 364-368.

Teimouri, M. (2010). Comparison of neural network and k-nearest neighbor methods in daily flow forecasting. Journal of Applied Sciences, 10.

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.

Tiedeman, C. R., Kernodle, J. M. and McAda, D. P. (1998). Application of nonlinear-regression methods to a ground-water flow model of the Albuquerque Basin, New Mexico: US Dept. of the Interior, US Geological Survey; Branch of Information Services [distributor].

Tiwari, M. K. and Chatterjee, C. (2010a). Development of an accurate and reliable hourly flood forecasting model using wavelet-bootstrap-ANN (WBANN) hybrid approach. Journal of Hydrology, 394(3), 458-470.

Tiwari, M. K. and Chatterjee, C. (2010b). Uncertainty assessment and ensemble flood forecasting using bootstrap based artificial neural networks (BANNs). Journal of Hydrology, 382(1), 20-33.

Tiwari, M. K., Song, K. Y., Chatterjee, C. and Gupta, M. M. (2013). Improving reliability of river flow forecasting using neural networks, wavelets and self-organising maps. Journal of Hydroinformatics, 15(2), 486-502.

Tongal, H. (2013). Nonlinear forecasting of stream flows using a chaotic approach and artificial neural networks. Earth Sciences Research Journal, 17(2), 119-126.

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).

Turan, M. E. and Yurdusev, M. A. (2009). River flow estimation from upstream flow records by artificial intelligence methods. Journal of Hydrology, 369(1), 71-77.

Tynchenko, V. S., Petrovsky, E. A., Tynchenko, V. V. and IEEE. (2016). The parallel genetic algorithm for construction of technological objects neural network models. 2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM).

Ulke, A., Ozkul, S. and Tayfur, G. (2011). Empirical methods for predicting suspended sediment load in Gediz River. Teknik Dergi, 22(2), 5387-5407.

Uzlu, E., Akpinar, A., Ozturk, H. T., Nacar, S. and Kankal, M. (2014). Estimates of hydroelectric generation using neural networks with the artificial bee colony algorithm for Turkey. Energy, 69, 638-647.

Vafakhah, M. (2013). Comparison of cokriging and adaptive neuro-fuzzy inference

system models for suspended sediment load forecasting. Arabian Journal of Geosciences, 6(8), 3003-3018.

Vasan, A. and Simonovic, S. P. (2010). Optimization of water distribution network design using differential evolution. Journal of Water Resources Planning and Management, 136(2), 279-287.

Vaze, J., Post, D. A., Chiew, F. H. S., Perraud, J. M., Teng, J. and Viney, N. R. (2011). Conceptual rainfall-runoff model performance with different spatial rainfall inputs. Journal of Hydrometeorology, 12(5), 1100-1112.

Wagener, T., Wheater, H. S. and Gupta, H. V. (2004). Rainfall-runoff modelling in gauged and ungauged catchments: World Scientific.

Wang, J., Wu, W., Li, Z. X. and Li, L. (2011a). Convergence of gradient method for double parallel feedforward neural network. International Journal of Numerical Analysis and Modeling, 8(3), 484-495.

Wang, J. H., Cai, Y. Q., Zhou, Y. L., Wang, R. L. and Li, C. W. (2011b). Discrete particle swarm optimization based on estimation of distribution for terminal assignment problems. Computers & Industrial Engineering, 60(4), 566-575.

Wang, J. Z., Liu, L. and Chen, J. (2008a). Combination of genetic algorithm and support vector machine for daily flow forecasting. Paper presented at the ICNC 2008: Fourth International Conference on Natural Computation, Vol 2, Proceedings.

Wang, K., Wang, X. D., Wang, J. S., Jiang, M. L., Lv, G. Y., Feng, G. L. and Xu, X. L. (2008b). Solving parameter identification problem of nonlinear systems using differential evolution algorithm. Paper presented at the Intelligent Information Technology Application, 2008. IITA'08. Second International Symposium on.

Wang, L., Zeng, Y. and Chen, T. (2015a). Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. Expert Systems with Applications, 42(2), 855-863.

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

Wang, Y., Guo, S. L., Xiong, L. H., Liu, P. and Liu, D. D. (2015b). Daily runoff forecasting model based on ANN and data preprocessing techniques. Water, 7(8), 4144-4160.

Wang, Y. H., Wang, H., Lei, X. H., Jiang, Y. Z. and Song, X. S. (2011c). Flood simulation using parallel genetic algorithm integrated wavelet neural networks. Neurocomputing, 74(17), 2734-2744.

Wang, Y. M., Kerh, T. and Traore, S. (2009b). Neural networks approaches for modelling river suspended sediment concentration due to tropical storms. Global Nest Journal, 11(4), 457-466.

Wang, Y. M. and Traore, S. (2009). Time-lagged recurrent network for forecasting episodic event suspended sediment load in typhoon prone area. International Journal of

Physical Sciences, 4(9), 519-528.

Weerts, A., Dogulu, N., López, P. L., Solomatine, D. and Shrestha, D. (2015). Estimation of predictive hydrologic uncertainty using the quatile regression and UNEEC methods and their comparison on contrasting catchments. Hydrology and Earth System Sciences, 19(7), 3181-3201.

Wei, C. C. (2016). Comparing single- and two-segment statistical models with a conceptual rainfall-runoff model for river streamflow prediction during typhoons. Environmental Modelling & Software, 85, 112-128.

Wei, S., Song, J. and Khan, N. I. (2012). Simulating and predicting river discharge time series using a wavelet-neural network hybrid modelling approach. Hydrological Processes, 26(2), 281-296.

Wei, S., Yang, H., Song, J., Abbaspour, K. and Xu, Z. (2013). A wavelet-neural network hybrid modelling approach for estimating and predicting river monthly flows. Hydrological Sciences Journal, 58(2), 374-389.

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.

Wu, C. L. and Chau, K. W. (2006). A flood forecasting neural network model with genetic algorithm. International Journal of Environment and Pollution, 28(3), 261-273.

Wu, C. L. and Chau, K. W. (2010). Using ANN-SSA coupled with UNEEC for uncertainty eetimate of daily rainfall-runoff transformation Hung Hom: Hong Kong Polytechnic Univ, Fac Construction & Land Use.

Wu, C. L. and Chau, K. W. (2011). Rainfall-runoff modeling using artificial neural network coupled with singular spectrum analysis. Journal of Hydrology, 399(3), 394-409.

Wu, C. L., Chau, K. W. and Fan, C. (2010). Prediction of rainfall time series using modular artificial neural networks coupled with data-preprocessing techniques. Journal of Hydrology, 389(1-2), 146-167.

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

Wu, C. L., Chau, K. W. and Li, Y. S. (2009b). Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques. Water Resources Research, 45(8).

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.

Wu, W. C. and Tsai, M. S. (2008). Feeder reconfiguration using binary coding particle swarm optimization. International Journal of Control Automation and Systems, 6(4), 488-494.

Xiong, L. H., O'Connor, K. M. and Guo, S. L. (2004). Comparison of three updating schemes using artificial neural network in flow forecasting. Hydrology and Earth System Sciences, 8(2), 247-255.

Xiong, T., Bao, Y., Hu, Z. and Chiong, R. (2015). Forecasting interval time series using a fully complex-valued RBF neural network with DPSO and PSO algorithms. Information Sciences, 305, 77-92.

Xu, J. W., Liu, Y. H., Zhao, J. F., Tang, T. and Xie, X. M. (2010). Improving TOPMODEL performance in rainfall-runoff simulating based on ANN. Paper presented at the 2010 International Conference on Display and Photonics.

Xu, J. W., Zhang, W. C. and Zhao, J. F. (2009a). Stream flow forecasting by artificial neural network and TOPMODEL in Baohe River basin. Paper presented at the IITAW: 2009 Third International Symposium on Intelligent Information Technology Applications Workshops.

Xu, J. W., Zhu, X. M., Zhang, W. C., Xu, X. X. and Xian, J. R. (2009b). Daily streamflow forecasting by artificial neural network ina large-scale basin. Paper presented at the 2009 IEEE Youth Conference on Information, Computing and Telecommunication, Proceedings.

Yang, R. F., Ding, J. and Liu, G. D. (1998). Preliminary study of hydrology-based artificial neural network. Journal of Hydraulics, 8, 23-27.

Yang, S. C., Lin, C. J., Lin, H. Y., Wang, J. G. and Yu, C. Y. (2016a). Image backlight compensation using recurrent functional neural network fuzzy network based on modified differential evolution. Iranian Journal of Fuzzy Systems, 13(6), 1-19.

Yang, W. A., Zhou, W., Liao, W. H. and Guo, Y. (2016b). Prediction of drill flank wear using ensemble of co-evolutionary particle swarm optimization based-selective neural network ensembles. Journal of Intelligent Manufacturing, 27(2), 343-361.

Yaseen, Z. M., Jaafar, O., Deo, R. C., Kisi, Ö., Adamowski, J., Quilty, J. and El-Shafie, A. (2016). Stream-flow forecasting using extreme learning machines: A case study in a semi-arid region in Iraq. Journal of Hydrology, 542, 603-614.

Yawson, D. K., Kongo, V. M. and Kachroo, R. K. (2005). Application of linear and nonlinear techniques in river flow forecasting in the Kilombero River basin, Tanzania. Hydrological Sciences Journal-Journal des Sciences Hydrologiques, 50(5), 783-796.

Ye, L., Zhou, J., Gupta, H. V., Zhang, H., Zeng, X. and Chen, L. (2016). Efficient estimation of flood forecast prediction intervals via single- and multi-objective versions of the LUBE method. Hydrological Processes, 30(15), 2703-2716.

Yigit, S., Eryigit, R. and Celebi, F. V. (2011). Optical gain model proposed with the use of artificial neural networks optimised by artificial bee colony algorithm. Optoelectronics and Advanced Materials-Rapid Communications, 5(9), 1026-1029.

Young, C. C., Liu, W. C. and Chung, C. E. (2015). Genetic algorithm and fuzzy neural networks combined with the hydrological modeling system for forecasting watershed runoff discharge. Neural Computing & Applications, 26(7), 1631-1643.

Yu, P. S., Chen, S. T. and Chang, I. F. (2006). Support vector regression for real-time flood stage forecasting. Journal of Hydrology, 328(3-4), 704-716.

Yu, P. S., Liu, C. L. and Lee, T. Y. (1994). Application of a transfer function model to a storage-runoff process Stochastic and Statistical Methods in Hydrology and Environmental Engineering (pp. 87-97): Springer.

Zahmatkesh, Z., Karamouz, M. and Nazif, S. (2015). Uncertainty based modeling of rainfall-runoff: combined differential evolution adaptive metropolis (DREAM) and K-means clustering. Advances in Water Resources, 83, 405-420.

Zaji, A. H. and Bonakdari, H. (2015). Efficient methods for prediction of velocity fields in open channel junctions based on the artifical neural network. Engineering Applications of Computational Fluid Mechanics, 9(1), 220-232.

Zecchin, A. C., Simpson, A. R., Maier, H. R., Marchi, A. and Nixon, J. B. (2012). Improved understanding of the searching behavior of ant colony optimization algorithms applied to the water distribution design problem. Water Resources Research, 48.

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.

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 50, 159-175.

Zhang, J. R., Zhang, J., Lok, T. M. and Lyu, M. R. (2007). A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. Applied Mathematics and Computation, 185(2), 1026-1037.

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

Zhang, Z. B., Jiang, Y. Z., Zhang, S. H., Geng, S. M., Wang, H. and Sang, G. Q. (2014). An adaptive particle swarm optimization algorithm for reservoir operation optimization. Applied Soft Computing, 18, 167-177.

Zhang, Z. L., Gao, C., Lu, Y. X., Liu, Y. X. and Liang, M. X. (2016). Multi-objective ant colony optimization based on the physarum-inspired mathematical model for bi-objective traveling salesman problems. Plos One, 11(1).

Zhao, D. M., Tang, J., Wu, X. G., Lin, C. G., Liu, L. J. and Chen, J. (2017). Developing a 2D vertical flow and sediment transport model for open channels using the Youngs-VOF method. Chinese Journal of Oceanology and Limnology, 35(2), 444-451.

Zhao, G. Q., Peng, Y. and Ma, X. L. (2011). A modified differential evolution algorithm and its application on neural network training. Paper presented at the 2011 International Conference on Intelligent Computation and Industrial Application, Sham Shui Po.

Zhao, H. Q., Qi, J. Q., Wand, J., Zheng, J. and Wu, W. (2010). Concentration estimation of gas mixture using a wavelet-based DPFNN. Chinese Journal of Sensors and Actuators, 23, 744-747.

Zhao, L., Hicks, F. E. and Fayek, A. R. (2015). Long lead forecasting of spring peak runoff using Mamdani-type fuzzy logic systems at Hay River, NWT. Canadian Journal of Civil Engineering, 42(9), 665-674.

Zhao, R. H. and Chen, S. Y. (2008). A hybrid fuzzy and neural network model for hydrological forecasting in ungauged basins. IAHS-AISH publication, 39-48.

Zhao, R. J. (1992). The Xinanjiang model applied in China. Journal of Hydrology, 135(1), 371-381.

Zhong, S. S. and Ding, G. (2005). Research on double parallel feedforward process neural networks and its application. Control and Decision, 20(7), 764-768.

Zhu, Y. M., Lu, X. X. and Zhou, Y. (2007). Suspended sediment flux modeling with artificial neural network: An example of the Longchuanjiang River in the Upper Yangtze Catchment, China. Geomorphology, 84(1), 111-125.

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.

Zounemat-Kermani, M., Kisi, Ö., Adamowski, J. and Ramezani-Charmahineh, A. (2016). Evaluation of data driven models for river suspended sediment concentration modeling. Journal of Hydrology, 535, 457-472.

Zounemat-Kermani, M., Kisi, Ö. and Rajaee, T. (2013). Performance of radial basis and LM-feed forward artificial neural networks for predicting daily watershed runoff. Applied Soft Computing, 13(12), 4633-4644.