A NEURAL NETWORK MODEL FOR FLOOD FORECASTING FOR SMALL HYDRO PLANTS

by

JIAN LI

B.Eng., Hohai University, China 1991

M.Eng., Hohai University, China 1994

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Civil Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

October 2005

Abstract

Artificial Neural Networks (ANNs) provide a quick and flexible way to create models for streamflow forecasting and have been shown to perform well in comparison with conventional hydrological models. This research applied multi-layer feedforward error backpropagation ANNs for real-time reservoir daily and hourly inflow forecasting. The proposed ANN models are trained by the Levenberg-Marquardt Backpropagation (LMBP) technique, coupled with an early stop method to avoid overfitting. A dataset partition method, which keeps the statistical properties of the training and the monitoring datasets as close as possible, is introduced to avoid under fitting. The method redistributes input/output patterns, in term of streamflow magnitude, into the training dataset and the monitoring dataset by breaking down the time series of the original data into subsets. The research introduced several indicators to cope with the snowmelt affected streamflow forecasting and overcome the limitation of snow information availability.

The performance of the daily time step ANN model is compared to an operational conceptual model (UBC Watershed Model) and a one time step lag model. The hourly time step ANN models are compared to a black-box model: Multi-Input Single Output Linear Model (MISOLM). The overall results of the research show that the ANN technique is practicable and effective for real-time streamflow and flood forecasting; the ANN models have higher simulation accuracy than the other referenced models.

The models developed have been implemented in BC Hydro. The real-time test of the models showed that ANN is a promising method for snowmelt affected streamflow and flood forecasting.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ANN	Artificial Neural Networks
ARIMA	Autoregression Integrated Moving Average
ARMA	Autoregression Moving Average
ARMAX	Autoregression Moving Average with Exogenous variables
BCH	BC Hydro
BP	Error Backpropagation
BPNN	Feedforward error Backpropagation Neural Network
CE	Nash-Sutcliffe Coefficient of Model Efficiency
CRR	Conceptual Rainfall-runoff model
FSM	Functional Series Model
GIUH	Geomorphologic Instantaneous Unit Hydrograph
LMBP	Levenbeg-Marquardt Backpropagation fast training algorithm
LMSE	Least Mean Squared Error
LPM	Seasonal based Linear Perturbation Model
MAE	Mean Absolute Error
MFSM	Muftuoglu form of Functional Series Model
MISOLM	Multi-Input Single Output Linear Model
MSE	Mean Squared Error
NAM	Nedbor-Afbtromrings-Model (Precipitation - Runoff Model)
NLP	Non-linear Prediction model
NNLPM	Nearest Neighbor Linear Perturbation Model
OSLM	One Step Lag Model ($Q_{t+1}=Q_t$)
R	Coefficient of Correlation
RBF	Radial Basis Function
RFS	River Forecast System
RMSE	Root of Mean Squared Error
SAC-SMA	Sacramento Soil Moisture Accounting model
SLM	Simple Linear Model

SOFM	Self Organizing Feature Map
SOLO	Self-organizing Linear Output Map
SORB	Self-Organizing Radial Basis Function ANN
SSNN	State Space Neural Networks
SWE	Snow Water Equivalent
UBCWM	UBC Watershed Model
UH	Unit Hydrograph
WATBAL	Water Balance model
WIFFS	Winnipeg Flow Forecasting System (a stochastic-deterministic watershed model)

Acknowledgements

The guidance and commitment of Dr. Ziad Shawwash as a friend, professor and advisor is gratefully acknowledged. The research could not have been done without his coordination between BC Hydro and UBC Civil Engineering. His excellent knowledge of optimization and his great contribution to BC Hydro made it possible for me to work in BC Hydro as a co-op student. His constant support and quiet attitude allowed completion of this research without external pressure or tension. Even during the absence caused by his illness, he still directed the research, on time, by email. Special thanks should be given to Dr. Thomas K. Siu. His commitment to learning while working made my first Canadian experience invaluable. He should also be recognized for his coordination of the research in BC Hydro and for his detailed explanation about BC Hydro and Hydro's power plant planning.

As my co-supervisor, Dr. Younes Alila deserved thanks for his detailed instruction in each phase of the research, for approving my research and for inspiring me through his passion for hydrology and his excellent teaching. I would also like to thank to Dr. Rob Millar for his support and encouragement.

The research could not have started without the support of Mr. Frank Weber and Ms. Stephanie Smith. They are warm-hearted hydrologists in BC Hydro. They provided all the hydrometric and climatic data for the research. Their clear explanations of hydrology, the case study watershed, the UBC Watershed Model, and flood forecasting in BC Hydro allowed me to get a good start with the research and make progress quickly. Frank also deserves the thanks for his efforts in implementing the research results (Artificial Neural Network model) in BC Hydro. His efforts made it possible to test the models with recent and real-time data. I also would like to thank Mr. Heiki Walk, Mr. Eric Weiss, Dr. Wuben Luo and Dr. Goran Srekvic of BC Hydro for their help and support.

My wife Meilin deserves special words for her love, support and time spent with my daughter. Thanks to my parents Zhisen and Xiuzhen for their love and encouragement every time I phoned them, even though we were separated by the Pacific Ocean.

CHAPTER 1 Introduction

1.1 Background

BC Hydro (BCH) is the third largest electric utility in Canada and its main activity is the production of electricity. Most of BCH production is used to meet the domestic load in BC, with some export and import to neighboring electricity markets in the US and Alberta. The capacity of the integrated hydroelectric system is about 11,500 MW, 87% of which is hydroelectric. Between 43,000 and 54,000 gigawatt-hours of energy is generated annually from 32 hydroelectric facilities and 2 thermal power plants.

BCH benefits from its hydroelectric power system by having a high degree of flexibility in its production activities. The economics of hydroelectric plants depend mainly on the reservoir's storage capacity, dam height and the magnitude and timing of inflows. Accurate forecasts of reservoir inflows enable BCH Planning, Scheduling & Operation Shift Engineers (PSOSE) to better manage and optimize the operation of their system and it helps them in making decisions on how to allocate the production of hydroelectric and thermal plants, on how much energy to buy or sell, and to control and regulate floods.

Currently, BCH uses the UBC Watershed Model (UBCWM), which is a conceptual hydrological model and has been integrated into BCH's River Forecast System (RFS), to forecast daily inflows to a number of reservoirs. The UBCWM requires an extensive knowledge of the geographical and the physiographical properties of a watershed, hydrological/thermal dynamics knowledge, and most important of all, the architecture of the model. To derive the inflow forecasts, BCH hydrologists and shift engineers need to be familiar with both basin-specific hydrological information and the model itself. The UBCWM calibration is usually a lengthy process that has to be repeated on regular basis.

In its current state, the RFS is only capable of generating daily inflow forecasts for the next five days. The first limitation of RFS is that it has only one model. It does not have an alternative way to forecast streamflow or to crossly verify forecast results. The second limitation of RFS is that the inflows are calculated as daily averages and not on an hourly basis. The lack of accurate representation of the hourly fluctuations of inflow within a day will often causes problems for real-time operation of reservoirs during flash flood events, particularly for small reservoirs. The third limitation of RFS is that it is very complex and can only be operated by hydrologists in the Hydrology & Technical Services at BCH, while the users of the forecast results, PSOSE, who are not hydrologists, may need to repeat the analysis frequently for daily and hourly inflow forecasting during rapidly changing situations.

The application of a commercial streamflow forecasting system is infeasible due to the time and the significant amount of money that would need to be spent on developing and adapting such a system. Many hydrological models and systems are currently used worldwide for streamflow and flood forecasting. These models have been developed mainly by research institutes, universities and government agencies. NWSRFS (NOAA, USA), MIKE FLOOD (DHI, Denmark) and EFFS (Delft, Netherlands) are some of the famous systems. It is well known that hydrological models are site specific, and most of hydrological models are developed for specific physiographic and hydrologic regimes in specific areas. Thus, commercial river flow forecast systems do not satisfy the needs of application in a specific area. The adaptation of an existing forecasting system to match a specific case would require extensive effort and knowledge of the watershed. The adaptation of a commercial streamflow forecasting tool to BCH would need a considerable amount of work.

To overcome the deficiencies of RFS, especially the second and third limitations, and to shorten the development period as much as possible, BCH's PSOSE initiated this research project with the intent to use an Artificial Neural Network (ANN) technique to check if it would be feasible and possible. There were three reasons that encouraged them to use ANN technique. Firstly, the

PSOSE group already uses an ANN technique to forecast hourly load and they are familiar with the technique and happy with its results. Secondly, our preliminary research on streamflow magnitude classification by using the Semi Supervised Support Vector Machine (S^3VM) technique, which is a kind of ANN technique, provided meaningful results. Lastly, BCH is licensed to use MetrixND that contains a powerful neural network tool box. MetrixND allows rapid development and implementation of ANN models and uses several fast training algorithms that can result in a rapid training and learning process.

1.2 Objectives of this Research

The overriding objective of this research project is to develop a flood forecasting tool that could be used to forecast floods at an hourly time step and meet the needs of PSOSE for their real-time and near-term operations. The tool could be based on ANN or any other promising technique, with the aim of eventually integrating it into the BCH's RFS. It was hoped that the methodologies investigated in the research would lead to the design of a flood forecasting tool that would be easy for PSOSE to understand and use. Inherent in this goal is the development of a prototype tool that will allow fast and accurate flood forecasts; and explore the best procedure to guide future application of the ANN technique to additional hydro-plants. Five sub-objectives were set for the research:

- Identify the most suitable flood forecasting technique and the architecture that is easy to understand, and fast in development, implementation and real-time streamflow forecasting;
- Develop a model which yields an acceptable operational accuracy for flood forecasting;
- Identify the main variables that have the largest effect on the accuracy of flood forecasts in the BCH system;
- Carry out a case study using the model, compare it with other potential techniques and present the results to BCH hydrologists and shift engineers for their review;
- Recommend how the model and/or tool could be integrated into an operational flash flood forecasting system at BCH.

1.3 Organization of the Thesis

Chapter 2 provides a literature review on flood forecast techniques with a special focus on the Artificial Neural Networks (ANN) technique. The literature review explores in detail the application of ANN in snowmelt runoff modelling. Chapter 3 provides the history of ANN and the mathematical background related to the ANN employed in this study. Chapter 4 provides a brief introduction to the case study watershed, data availability, and the criteria used to judge model performance. It also illustrates the applications of the ANN technique in a case study watershed for daily streamflow forecasting, and compares the ANN models performance with the UBC Watershed Model (UBCWM). The chapter also illustrates an application of ANN for hourly streamflow forecasting, and it compares the ANN performance with the Multi-input Single Output Linear Model (MISOLM). Chapter 5 provides the conclusions and recommendations for future research.

CHAPTER 2 Literature Review

2.1 Conventional Flood Forecasting Technique

Streamflow forecasting, which is a special field of hydrology, can be divided into two categories, namely storm event forecasting and continuous streamflow forecasting. Storm event forecasting focuses on the peak discharge, total volume of flood flow, or discharge temporal distribution of a single storm event. The continuous streamflow forecasting predicts the streamflows over time.

Most streamflow forecasting methods rely on hydrological models. Storm event forecasting involves the event models, and continuous streamflow forecasting uses continuous models. Both event and continuous models are deterministic models, in which a given input of precipitation must produce a fixed output of runoff in a certain physical setting: its output is determined by the input (Jones, 1997). The deterministic models could be black-box, conceptual, or physically-based models. Jones (1997) summarized the classification of hydrological models as shown in Figure 2.1. In terms of model spatial discretization, lumped models, which include semi-distributed conceptual and black-box models, have been widely used in real-time streamflow forecasting, while physically based distributed models are rarely used in real-time flood forecasting.

Black-box models have long been used as efficient flood forecasting tools. These models attempt to predict overall catchment response and the process involved is based on the interpretation of the observed response of a catchment (Sivapalan et al., 2003). The models are generally derived from the analysis of historical data. Sherman devised an extremely valuable method, Unit Hydrograph (UH), for determining the flood hydrograph in 1932. The Sherman UH and its later variants are widely used and may be found at the heart of many modern and highly sophisticated computer

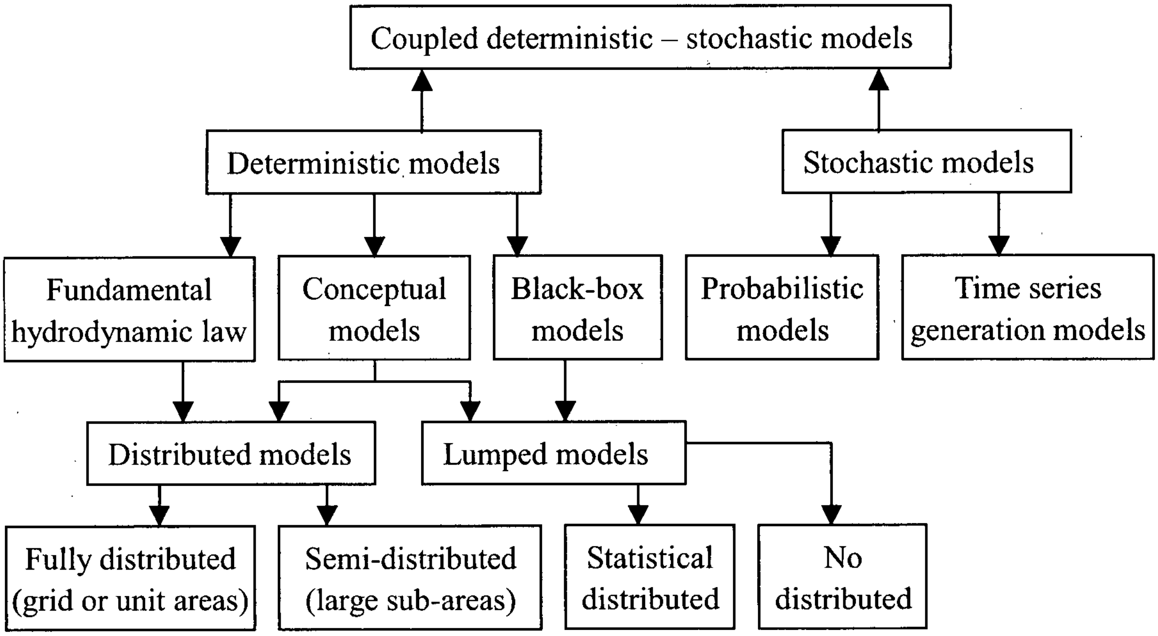
Coupled deterministic – stochastic models
Deterministic models
Stochastic models
Fundamental hydrodynamic law
Conceptual models
Black-box models
Probabilistic models
Time series generation models
Distributed models
Lumped models
Fully distributed (grid or unit areas)
Semi-distributed (large sub-areas)
Statistical distributed
No distributed

Figure 2.1 The classification of hydrological models (Jones, 1997)

models (Jones, 1997). The Sherman UH variants include Instantaneous Unit Hydrograph (IUH), Synthetic Unit Hydrograph (SUH), Geomorphic Instantaneous Unit Hydrograph (GIUH) and Nash Unit Hydrograph. The Unit Hydrograph approach is based on linear and time-invariant assumption of rainfall-runoff processes. Since Sherman introduced the concept of UH, linear system analysis has played an important role in applied hydrology, in rainfall-runoff modelling and in flood routing. Black-box models that fall in this category include Simple Linear Model (Nash and Foley, 1982), Linear Perturbation Model (Nash and Barsi, 1983), Multiple Input/Single Output Linear Model (Kachroo and Liang, 1992), Nearest Neighbour Linear Perturbation Model (Shamseldin and O'connor, 1996). The application of black-box models also involves, in modelling terminology, model calibration. Black-box model calibration is generally a process of solving a matrix equation by least square method, and it also uses optimization techniques for specific types of black-box models. A more systematic way to develop black-box model can be referred to the downward approach. It focuses on determining input-output relationships from datasets in an efficient manner with little interest in what is happening within the system and why this is happening (Sivapalan et al., 2003). The approach starts from analyzing relationship between input and output, then refines the model by increasing model complexity to further interpret input-output relationships. The model structure is inferred from data rather than being preconceived from physical processes. Young (2002) developed a stochastic downward approach, called data-based mechanistic approach, and consequently a low-order nonlinear model, to modelling streamflow with limited set of rainfall-runoff data. The downward approach may fully take advantage of a given set of information. Therefore, the derived model should have good performance in real-time streamflow forecasting.

Conceptual models try to represent rainfall-runoff processes in a simplified way including empirically calculated components. Compared with black-box models, conceptual models are more complex. These models produce a single output, the streamflow, but usually do not simulate other hydrological variables like infiltration and groundwater level (Institute of Hydrology Modelling Group, 1999). The models in this category include Xinanjiang model (Zhao, 1984), Sacramento Model (Burnash, 1995), TANK model (Sugawara, 1995), UBC Watershed Model (Quick, 1995), TOPMODEL (Beven, 2000). Conceptual models employ parameters to represent certain part of the rainfall-runoff processes. Some parameters are predetermined by either a physical theory or watershed characteristics, while other parameters need to be calibrated and/or optimized. The parameter optimization, either automatic or empirical optimization, tries to make simulated streamflow match observed streamflow as close as possible by interactively altering parameters. Inevitably, optimization could contain potential pitfalls: (1) different optimization methods may identify different set of parameter; (2) different people may end up with different optimal parameter sets; (3) skill is required in selecting initial value and suitable range for each parameter. The optimized parameters become merely grist to the calibration mill and may compensate for model structure error, resulting in an inaccurate representation of the catchment processes (Sivapalan et al., 2003). Recent debates on conceptual models argue that although there are numerous conceptual models around and being developed along with significant improvements in model calibration, we are still nowhere near solving the problems related to arbitrary model structure and a priori estimation of parameters, that hamper predictions (Sivapalan et al., 2003). Conceptual models dominated most of the research conducted during the last century, but failed to solve many problems of identifying, estimating and validating models (Young, 2002).

The physically-based distribution models use 'theoretical' equations for rainfall-runoff related processes. For example, the models use the Richard equation for water movement in unsaturated zones, St. Venant equation for streamflow routing and the Kinematics wave equation (simplified St. Venant equation) for hillslope runoff routing. The distributed models use spatially distributed parameters and variables to deal with the spatial heterogeneity of watershed and spatial variability of input information. The distributed models can simulate many hydrological components (e.g. surface runoff, groundwater level) and analyze the hydrological impact of many factors such as land use changes. Due to the model complexity and relatively high number of parameters at each grid, it is impractical to use distributed model in real-time streamflow forecasting.

2.2 Artificial Neural Networks (ANN) Technique

Artificial Neural Networks (ANN) are well known, massively parallel computational tools that have exhibited excellent performance in solving various complex and highly non-linear science and engineering problems due to their ability to recognize relationships between model input(s) and output(s). Although the prototype ANN was introduced in the early 1940s, it was not until early 1980s did it begin to experience a resurgence of study and application. The resurgence was mainly caused by the rediscovery of a mathematically rigorous theoretical algorithm: the error backpropagation learning rule (Rumelhart et al., 1986) for a multilayer ANN training, and the emergence of powerful digital computational tools. Since the early 1990's, ANNs have been successfully used in hydrology-related areas such as rainfall-runoff modelling, streamflow forecasting, ground-water modelling, water quality, water management policy, precipitation forecasting, hydrologic time series, and reservoir operations (ASCE Task Committee, 2000 I). A comprehensive review of the application of ANNs in hydrology can be found in ASCE Task Committee (2000 I and II).

Almost all researchers show that ANN models are superior to conceptual hydrological models such as SAC-SMA (Gupta et al., 1997), PREVIS (Coulibaly et al., 2000), Storage routing model (Xu and Li, 2002), Tank and NAM (Tingsanchali and Gautam, 2000), CRR model (Jain et al., 2004) and GIUH (Zhang and Govindaraju, 2003) in the way of modelling accuracy or efficiency statistics such as Root Mean Square Error, Nash-Sutcliffe Coefficient of Model Efficiency, Coefficient of Correlation, volume error etc.

ANNs are also more tolerant to noisy input information and are more stable and consistent than other types of black-box model like ARMA, ARMAX (Nayak et al., 2004; Chang and Chen, 2001; Coulibaly et al., 2000; Chang et al., 2004; Xu and Li, 2002; Tingshanchali and Gautam, 2000), ARIMA (Chang et al., 2002), SLM, LPM, NNLPM (Shamseldin, 1997; Rajurkar et al., 2004), NLP (Laio et al., 2003) and FSM, MSM (Sajikumar and Thandaveswara, 1999).

While most researches are focused on rainfall-runoff modelling, some researches are focused on snowmelt streamflow simulation (Coulibaly et al., 2000; Zealand et al.,1999; Tokar and Johnson, 1999). There are also small number of researchers who are focused on flood peak forecasting (Smith and Eli, 1995) and rainfall-runoff mechanism analysis (Gautam et al., 2000).

All researches take a case study watershed as a lumped unit. Many watersheds of varying size and climate have been studied. ANNs have been applied in watersheds that vary from as small as 0.5255km^2 (Gautam et al., 2000) to more than thousands of square kilometers (26200km^2, Rajurkar et al., 2004; 19270km^2, Zealand et al., 1999; 18000km^2, Shamseldin, 1997) all around the world. The time step for the streamflow forecasting varies from 10 minutes (Gautam et al., 2000) to one month (Sajikumar and Thandaveswana, 1999). All of these applications have proved that ANNs are adaptable and versatile.

There are three types of ANNs that are used in hydrological modelling: the three-layer feedforward error Backpropagation Neural Network (BPNN), the Radial Basis Function (RBF) and the Recurrent Neural Network (RNN). The most frequently used ANN models are BPNN model (Rajurkar et al., 2004; Kim and Barros, 2001; Gautam et al., 2000; Coulibaly et al., 2000; Zealand et al., 1999; Shamseldin, 1997; Jain et al., 2004; Sudheer et al., 2003; Xu and Li, 2002; Tingsanchali and Gautam, 2000; Laio et al., 2003). BPNN is characterized by conceptually simple algorithm, enough capacity for practical problems and lots of existing implementation tools. In recent years, more and more researches began to use the RBF (Moradkhani et al., 2004; Lin and Chen, 2004; Chang and Chen, 2001) and the RNN (Zhang and Govindaraju, 2003; Chang et al., 2004). The RBF model is easy to build and train due to the number of hidden nodes that can be determined during the training process. RBF training avoids the empirical trial-and-error processes in BPNN architecture selection. The single most drawback of RBF is that it tends to use a lots of hidden neurons, which means that RBF tends to memorize input patterns and therefore looses the generalization ability. For RNN, its feedback feature is analogous to the time delay that is usually encountered in rainfall-runoff processes. By using RNN, researchers don't need to use the trial and error method to determine how many rainfall inputs are lagged. Therefore, RNN generally uses much less number of inputs than BPNN, and consequently its architecture is simple. The shortcoming of RNN is that the RNN training is relatively difficult to pursue due to its feedback features, as RNN should be trained at a very low learning rate to keep the training process stable.

Recent researches tend to try new types of ANN architecture, such as hybrid ANNs. These new architectures include Self-organization Radial Basis Function ANN (Moradkhani et al., 2004), Neuro-fuzzy Network (Nayak et al., 2004), and Dynamic Feedback/Recurrent Network (Chiang et al., 2004).

The application of ANNs in streamflow modelling can be categorized into two groups, non-updating mode and updating mode. The updating mode uses previous observed streamflow as model input while non-updating mode doesn't. The ANN models, which run in non-updating mode, are a kind of rainfall-runoff model. It is obvious that the models run in updating mode should have higher modelling accuracy than those run in non-updating mode due to the high autocorrelation (ACF) of streamflows. The ability of using recent streamflow in modelling is one of the basic advantages of ANN models. Actually, ANN models can accept any kinds of input information, both quantitative and qualitative.

2.3 The Application of ANN in Snowmelt Runoff Modelling

This literature review found that the research on the use of ANN for snowmelt runoff modelling is limited and that the existing researches have shown the robustness of ANN in this area. One of the earliest publications (Tokar and Johnson, 1999) in this area showed that daily ANN model that uses daily precipitation, temperature and snow water equivalent (SWE) as inputs provides a high degree of accuracy when compared with a regression model and with a simple conceptual model. Their results indicated that temperature is an important indicator for snowmelt runoff simulation. As summarized by ASCE task committee, ANN models have equal or better performance than conceptual hydrological models, like the WATLAB and the SAC-SMA models, particularly in snow affected streamflow simulations.

Zealand et al. (1999) compared standard Feedforward Backpropagation Network with the conceptual Winnipeg Flow Forecasting System (WIFFS) in Namakan Lake sub-watershed (19,270km^2) of the Winnipeg River Basin (Canada) for sub-monthly (weekly) streamflow forecasting. In the neural network model, past and forecasted total weekly precipitation, average weekly temperature and past weekly streamflow were used as inputs. When the two models were compared fairly and unbiasly using the same input information, ANN performance was marginally better than that of the WIFFS model. They optimized the ANN inputs by adding "period of year" and the accumulative precipitation, and deleting relatively insignificant input by sensitivity analysis. The optimized ANN model for 1, 2, 3 and 4 week ahead forecasting were found to perform better than those reached using the WIFFS model. The research also indicated that improvements in performance of the ANN model in simulation were only marginal as training time increases from 10 minutes to 2 hours. Coulibaly et al. (2000) compared the performance of standard multilayer feedforward neural network with PREVIS, which is a conceptual model, and autoregressive moving average with exogenous inputs (ARMAX) model for daily reservoir inflow in the Chute-du-Diable watershed (9700km^2) in North Quebec, Canada. The ANN model used forecasted maximum, minimum and mean temperature, precipitation, snowmelt and past reservoir inflow and precipitation to forecast one day ahead reservoir inflow. The common trial-and-error method was employed to select the number of hidden neurons. Early stop technique and LMBP were used to train the ANN. The research indicated that ANN has a substantially better prediction accuracy than the other two models; and a low deterioration of prediction performance with the increase in the forecast lead-time. The research also showed that the early stop technique provides faster training than training to complete convergence and it also secures the network against overfitting.

2.4 Recent Progress in Application of ANN in Hydrology

Researches on application of various ANNs (BPANN, RBF, recurrent ANN and cascade correlation algorithm) have proven that ANNs are robust modelling tools for the complex, highly non-linear and somewhat difficult to model hydrological processes. ANNs were also found to be superior to both conceptual hydrological models and black-box models. In the last two years, researches in this field have moved to a new phase that is characterized by hybrid ANNs, ANN architecture comparison and the effort to relate ANNs to physical processes.

2.4.1 Hybrid ANN Model

Moradkhani et al. (2004) built a hybrid ANN model (SORB) by combining two ANN architectures: Self Organizing Feature Map (SOFM) and Radial Basis Function (RBF), for one-step ahead streamflow forecasting. The key feature of SORB compared with the original RBF is that SORB uses SOFM to get all RBF parameters, which include node center and the Gaussian function parameter. RBF can be considered as a three-layer network in which each hidden node is thought of as a center of a category of input patterns. The RBF calculates the similarities between input patterns in each hidden node and fires hidden nodes in different degree by an activation function.

The weighted sum of hidden nodes outputs will form the RBF output. The number of centers (hidden nodes) of RBF are generally chosen by the researcher or by using a k-mean clustering technique. In SORB, the number of centers and parameters of RBF are derived from SOFM. SOFM is used to classify the input patterns into several typical or representative patterns. The representative patterns are identified by pattern center and standard deviation and, then, they are adapted by RBF of SORB. SORB was used to develop a one-step ahead daily streamflow forecast model for the Salt River (10,000km^2), which is a sub-watershed of the lower Colorado River. The authors (Moradkhani et al., 2004) concluded that SOFB is superior to linear regression model, the BPNN and the Self-Organizing Linear Output Map (SOLO).

Rajurkar et al. (2004) developed a non-updating hybrid ANN model by coupling BPNN with a Simple Linear Model (SLM) which imitates Unit Hydrograph theory and uses a response function to calculate runoff from rainfall. In the ANN model, runoff is first estimated by the SLM according to rainfall, then refined by adding rainfall as another input to the ANN with the idea that ANN can handle non-linear component in rainfall-runoff processes. The application of the model in six catchments worldwide showed that the approach is very useful in modelling the rainfall-runoff relationship in the non-updating mode.

ANNs and fuzzy logic approaches have been proven to be efficient when applied individually. The growing interest in combining them have lead to a successful application of neural-fuzzy architecture in signal processing and related areas. Fuzzy models are a structured quantitative estimator where the system behavior is described by natural language. Fuzzy architecture combines rule based systems and fuzzy control algorithms to describe complex non-linear processes. Chang and Chen (2001) introduced fuzzy concept into RBF to explain its behavior. The behavior of hidden nodes, which calculates the similarity between input patterns and the center of a category of patterns (typical patterns), can be represented by the 'IF' statement in a fuzzy rule; the connections between hidden layer and output layer represent the 'THEN' part of a rule. Nayak et al. (2004) applied Adaptive Neuro-Fuzzy Inference System (ANFIS) in Baitarani River, India for daily streamflow modelling. The neuro-fuzzy model was coupled with fuzzy inference and was trained by a method which is a combination of gradient descent and least square estimate method. The results indicated that ANFIS forecasted streamflow preserved the statistical properties of the original streamflow and had good performance in terms of various statistical indices. The author also showed that: (1) ANFIS performance is better than ARMA but similar to BPNN; (2) transforming inputs to normally distributed time series by Wilson-Hilferty transformation improves ANFIS performance.

2.4.2 Physical Processes Inherent in ANN Model

After significant success in the applications of ANN in various aspects of hydrology, the motivation to get rid of the black-box nature of ANN and explain ANN in the hydrology terminology starts to emerge. Recent researches were focused on relating ANN behavior to physical processes, relating ANN architecture to existing hydrological theory or even determining the ANN architecture based on geomorphologic information.

Zhang and Govindaraju (2003) demonstrated how geomorphologic information could be incorporated into ANN architecture. In their research, Geomorphologic Instantaneous Unit Hydrograph (GIUH) provided a theoretical basis for part of the architecture of Geomorphologic-based ANN (GANNs). They interpreted the number of hidden nodes of a three-layer BPNN by the number of possible paths in GIUH and therefore, it can be determined by geomorphologic information. The weights of connections between hidden layer and output layer are represented by path probability in GIUH theory and determined by geomorphologic information. The weights of connections between input layer and hidden layer are analogous to the transient holding time when compared with GIUH theory. The inputs to GANN are current and past rainfall excess. The number of previous rainfall excesses data points was determined by trial-and-error method. The application of GANN in two small watersheds in southern Indiana showed that the GANN is superior to GIUH. What is meaningful in the research is that the GANN has changed the black-box nature of ANN, as it evaluates ANN from a purely empirical model to, somewhat, physically based model.

Pan and Wang (2004) derived UH from weights of network. They employed dynamic recurrent neural networks (also: State Space Neural Networks, SSNN) for hourly time step flood forecasting in Wu-Tu watershed (204km^2) in Taiwan. The research built connection between physical concepts and ANN parameters by deriving UH from weights of network. The research revealed the physical concept in networks, which is traditionally thought to be a black-box model as it lacks any physical meaning of the weights. The performance of SSNN for short term rainfall-runoff forecasting indicated that the dynamic recurrent network is an appropriate tool.

Jain et al. (2004) showed that ANN is able to capture certain physical behavior of rainfall-runoff processes. They applied a three-layer BPNN with four hidden nodes in Kentucky River Basin (10239km^2, USA) to simulate five hydrological processes simultaneously and compared the ANN model with the Conceptual Rainfall Runoff model (CRR). The CRR model consists of two components and has the ability to simulate total flow, base flow, surface flow, soil moisture and infiltration processes simultaneously. The comparison between ANN behavior (hidden node output) and output of CRR model was done by means of correlation analysis and graphical comparison. The comparison indicated that both the ANN model and CRR are good and the performance of the ANN model is better than that of CRR model. The ANN model captures different components of the physical rainfall-runoff process being modelled through its massively distributed structure. Those four hidden nodes in the ANN model were found to model different components of a hydrological process individually. The inference from the study strongly suggests that an ANN model is not necessarily a black-box model as it is able to capture certain physical behavior of rainfall-runoff processes.

CHAPTER 3 Introduction to ANN

An ANN is an information processing system that has certain performance characteristics in common with biological neural networks. ANNs have been developed as generalizations of mathematical models of human cognition or neural biology (Fausett, 1994). For engineering problems, neural networks are a powerful tool for modelling in which the explicit form of the relationship among inputs and outputs are unknown.

3.1 History of ANN

The development of ANN dates back to 1943, motivated by a desire to try both to understand the brain and to emulate some of its strengths. Fausett (1994) indicated that Warren McCulloch, a neurophysiologist, and Walter Pitts, a mathematician, modelled a simple neural network with electrical circuits and proposed the notion of threshold logic units. Their model, the McCulloch-Pitts neuron, is generally regarded as the first neural network. The threshold logic means that if the weighted sum of inputs to a neuron is greater than the threshold then the unit will be fired (Figure 3.1).

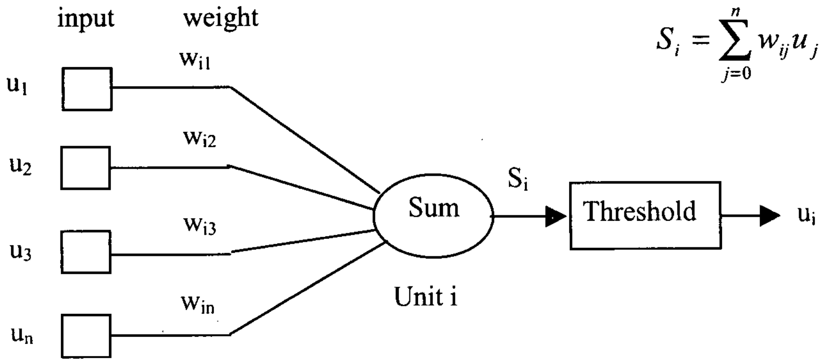
$$S_i = \sum_{j=0}^{n} w_{ij} u_j$$

Figure 3.1 Diagram of McCulloch-Pitts neuron

The 1950s and 1960s were the first golden age of ANN. The period is characterized by the introduction of several neural networks learning rules. The famous one is the delta rule, which adjusts the weights to reduce the difference between the neural network output and desired output in terms of Least Mean Squared Error (LMSE). The delta learning rule for a single-layer neural network is a precursor of the error backpropagation learning rule for multilayer net.

The 1970s were a quiet period in ANN history. There were two reasons that have caused the disinterest. The first one was the failure of single layer perceptron network, while the second one was the lack of a general method of training a multilayer net.

The early 1980s saw resurgence of neural network sparked off by Hopfield. Hopfield developed a network called content addressable memories that can store several patterns, like human face, and recall successfully stored patterns by whole or partial pattern. His studies showed the power of ANN. Other milestones in this period were the Boltzmann machine and backpropagation training algorithm (BP). The BP idea of propagating information about error at the output units back to the hidden units was first discovered in 1974. It was rediscovered and popularized by Rumelhart, Hinton and Williams in1986. The algorithm is a generalized delta rule.

Recent interest in ANN can be attributed to two reasons. The first, is the training technique that has been developed for more sophisticated network architecture and is able to overcome the shortcomings of the early simple neural network. The second, is high-speed digital computer that makes the training and simulation of neural processes more feasible.

3.2 A Brief Introduction to ANN

3.2.1 Introduction

An ANN is a massively parallel-distributed information processing system that has certain performance characteristics resembling biological neural networks of the human brain (Haykin, 1994). A neural network is characterized by its architecture that represents the pattern of connection between nodes, its method of determining the connection weights, and the activation function (Fausett, 1994). Like the human brain, a typical ANN consists of a set of nodes (neurons), each generates an output value or activation determined by the sum of the inputs to the node. Nodes are then organized as a layer. Several layers connected one to another form an ANN. The first layer is called the input layer and the last layer is the output layer. The layers in between the input and the output layers are hidden layers. ANN can be classified into three classes according to the number of layers: single layer neural net (perceptron), bi-layer neural net and multi-layer neural net (multilayer perceptron).

The multilayer perceptron is the most common and useful neural net. The number of nodes in each layer is not fixed. The number of nodes in the input and the output layers are problem specific while the number of nodes in a hidden layer is determined by a trial-and-error procedure. The nodes in neighboring layers are fully connected by links. The connection between two nodes is weighted so that the significance of each input to a node is either emphasized or inhibited. It is the weights that determine the characteristics of the node. Different network models can be made to change their behavior to fit for different applications by modification of weights. The arrangement of neurons into layers and the connection patterns within and between layers is called the net architecture. Two basic paradigms for neural networks can be observed: feedforward net (Figure 3.2) and recurrent net (Figure 3.3).

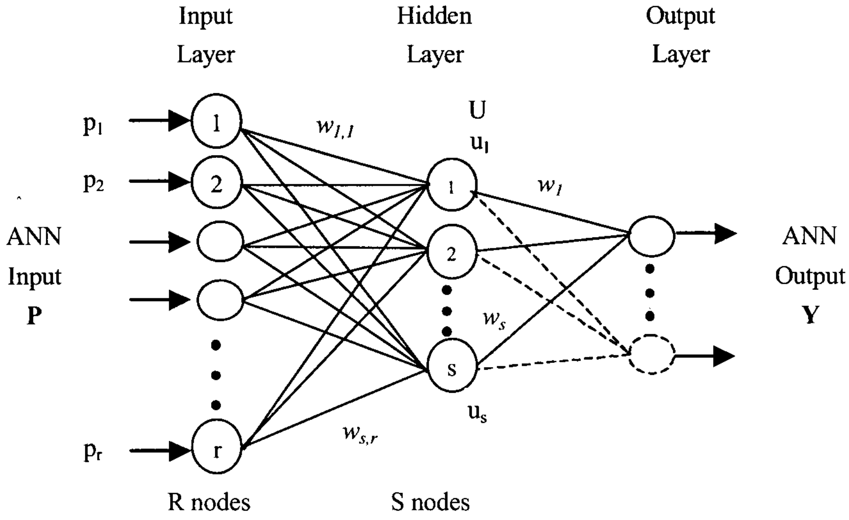

Figure 3.2 Architecture of feedforward neural network

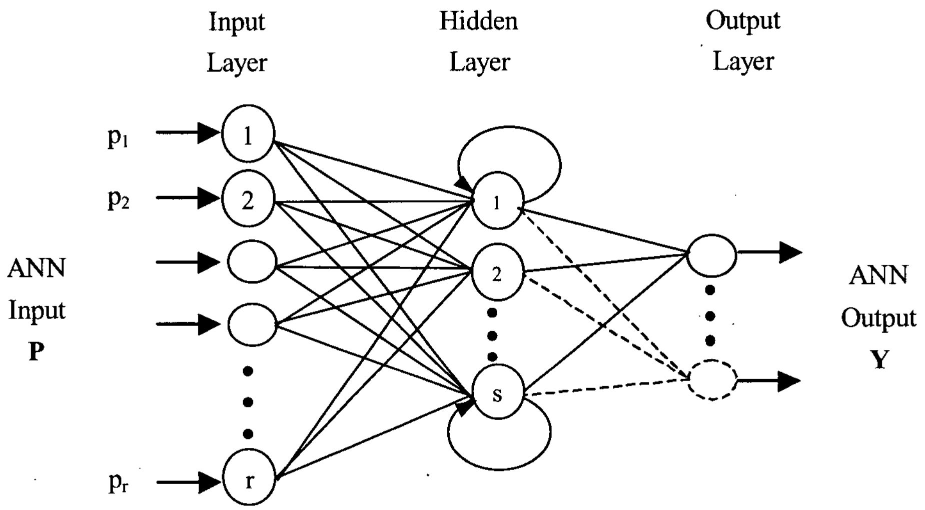

Figure 3.3 Architecture of recurrent neural network

Figure 3.2 shows a three layer feedforward neural net model. In feedforward net, information is passed from input layer to hidden layer(s) then output layer, or say, output of one node is fed to another layer of nodes. The input layer receives the input variables (or input vector P, or input pattern) from the research problem. The input vector may consist of quantified information that is thought to affect the output. Vector Y is the neural net output vector that consists of values simulated or predicted by the neural net. The single most difference between feedforward and recurrent neural nets is that in recurrent networks (Figure 3.3), the output of a node is an input of both other nodes and itself, thus allowing feedback.

Multilayer nets can learn any continuous mapping to any arbitrary accuracy. Kolmogorov mapping neural network existence theorem states that a feedforward neural network with three layers of nodes (input, hidden and output nodes) can represent any continuous function exactly (Fausett 1994, Charpter 6.1). Furthermore, for a neural net with bias terms, a sigmoid layer and a linear output layer are capable of approximating any function with a finite number of discontinuities (Demuth and Beale, 2004). The three-layer feedforward ANN has been the most successful for practical applications in various fields and is the most commonly used in practice.

3.2.2 Algorithm of Feedforward ANN

The basic element of an ANN is nodes (neurons), which include input nodes, hidden nodes and output nodes in corresponding layers. The input nodes do nothing but pass input information p_i to hidden nodes. Hidden nodes and output node(s) calculate the weighted sum S_i of inputs to itself, then transfer the weighted sum by an activation function $f(S_i)$ and give node output u_i=$f(S_i)$. In Figure 3.2, suppose there is only one output node, $P=(p_1, p_2, \cdots, p_r)'$ is input vector, $U=(u_1, u_2, \cdots, u_s)'$ is output vector by hidden nodes, W_{12} is weight matrix representing the weight $w_{s,r}$ which connects the node s in hidden layer and node r in input layer. The output of hidden layer is:

$$U = f(W_{12}P + B_h) \quad \text{or} \tag{3.1}$$

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_s \end{bmatrix} = f\left(\begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,r} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,r} \\ \\ w_{s,1} & w_{s,2} & \cdots & w_{s,r} \end{pmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_r \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_s \end{bmatrix}\right) \tag{3.2}$$

where $B_h = (b_1, b_2, \cdots, b_s)'$ is bias term for hidden nodes. With the hidden nodes output on hand, the ANN output is:

$$Y = f(W_{23}U + B_o) \quad \text{or} \tag{3.3}$$

$$Y = f\left(\begin{bmatrix} w_1 & w_2 & \cdots & w_s \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_s \end{bmatrix} + B_o\right) \tag{3.4}$$

where $W_{23} = (w_1, w_2, \cdots, w_s)$ is weight vector representing the weights between hidden nodes and output node (one node in output layer); B_o is bias term associated with output nodes.

The function $f(.)$ is an activation function. In practical situation, same activation functions are used for all nodes in a layer, although it is not required. Generally, hidden nodes use nonlinear activation function to take advantage of a multilayer net. The form of an activation function determines how a node responds to the total input information it receives. The most commonly used activation function is the sigmoid function. They take the form of binary sigmoid function as shown in Figure 3.4:

$$f(x) = \frac{1}{1+e^{-x}} \tag{3.5}$$

or bipolar sigmoid function as shown in Figure 3.5:

$$f(x) = \frac{2}{1+e^{-x}} - 1 \tag{3.6}$$

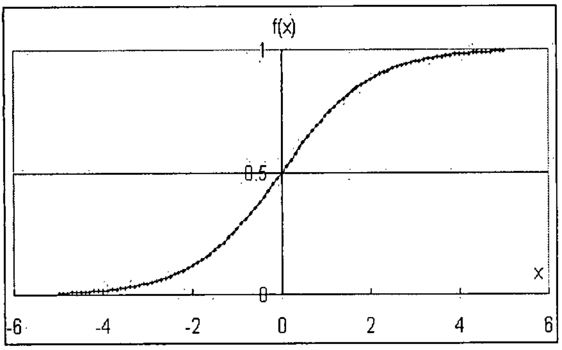

Figure 3.4 Binary sigmoid function

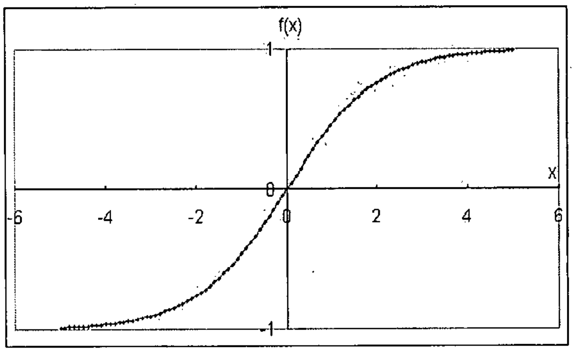

Figure 3.5 Bipolar sigmoid function

Linear activation function or identity function *f(x)=x* could also apply to input nodes and output nodes. The sigmoid function is a bounded, monotonic, nondecreasing function that provides simple relationship between the value of the function at a point and the value of the derivative at the same point. This dramatically reduces the burden during network training.

3.3 Error Backpropagation for Neural Network Training

3.3.1 The Types of Training

Broadly speaking, there are two categories of training, supervised and unsupervised training. Supervised training can be called ANN model calibration in the hydrologic terminology. In supervised training, both the input patterns and target outputs are provided. The training is accomplished by presenting a sequence of training patterns, and comparing their resulting output against the target output. Weights of an ANN are then adjusted to minimize the difference between the calculated and targeted outputs according to certain learning algorithms. In unsupervised training, a network is provided with input patterns but without target output patterns. The network itself then decides what features it should use to group the similar input patterns together. The typical member of each group does not need to be specified. The neural net will produce a representative pattern, which is the center of the group of patterns, for each cluster formed. This is often referred to as self-organization or clustering.

3.3.2 The Error Backpropagation Training Algorithm

The error backpropagation (BP) training algorithm is one of the most important and widely used techniques for neural network training. The BP algorithm makes it easy to find the network error gradient for a given pattern. It was created by generalizing the Widrow-Hoff learning rule to multilayer network and non-linear differentiable transfer function. Actually, the BP algorithm is used almost exclusively with feedforward multilayer neural network using continuously valued nodes. Based upon the Mean Squared Error and the Gradient Decent methods, the BP algorithm uses input vectors and corresponding output vectors to train a network until it could approximate a function (i.e. associate input vectors with specific output vectors). The BP algorithm involves two steps. In the first step, input information goes forward from input layer to hidden layer(s) then reaches the output layer. The error signal at the output layer could be obtained easily by comparing ANN output with target output. In the second step, the error signal at the output layer is propagated backward to the hidden layer(s) and input layer, weights are adjusted according to the error signal of related nodes.

The BP learning algorithm aims at finding the optimal weight matrices and bias vector that minimize the total error generated by all patterns in the training dataset. The total error always takes the form of squared difference between desired and actual output:

$$E = \frac{1}{2}\sum_{t=1}^{P}\sum_{j=1}^{N}\left(u_{t,j} - C_{t,j}\right)^2 \tag{3.7}$$

where P is the number of training patterns, N is the number of nodes in output layer, u is the ANN output at the output layer, and C is the desired output. When neural net has only one output neuron, the error function is simplified to:

$$E = \frac{1}{2}\sum_{t=1}^{P}(u_t - C_t)^2 \tag{3.8}$$

The basic BP learning algorithm updates network weights and bias in a direction in which the total error decreases most rapidly along the gradient:

$$\Delta_p w_{i,j} = \alpha \frac{\partial E^p}{\partial w_{i,j}} \tag{3.9}$$

where α is learning rate, $w_{i,j}$ is weight of the connection between node i and j in neighboring two layers, node j is located in former layer, while node i is located in later layer. The updated network weights, $w^*_{i,j}$, can be expressed as:

$$w^*_{i,j} = w_{i,j} + \alpha\delta_i u_j \tag{3.10}$$

where δ_i is error signal, when a node is an output node:

$$\delta_i = u_i(1 - u_i)(C_i - u_i) \tag{3.11}$$

or when a node is a hidden node:

$$\delta_i = u_i(1 - u_i)\sum_h \delta_h w_{h,i} \tag{3.12}$$

where h is the number of nodes in the output layer.

The standard BP algorithm is a gradient descent algorithm, as is the delta rule. There are a number of variations on the basic algorithm which are based on other standard optimization techniques such as the Conjugate Gradient method, the Quasi-Newton method and the LMBP (a more general training method). Quasi-Newton and standard BP learning algorithms are two special situation of LMBP. Matlab Neural Net Toolbox has implemented a number of these variations. In practice, the LMBP is faster and finds better optima for a variety of problems than other methods (Demuth and Beale, 2004).

The main problem associated with BP algorithm is that it is not guaranteed to find a global minimum. In fact, once a minimum has been reached, it is difficult to determine whether it is a global minimum or not. An effective way to avoid this problem is training the ANN with several sets of initial weights and biases, then choose the best result. The BP algorithm does not have the ability to incrementally learn from training patterns, therefore preventing the incremental accumulation of knowledge.

3.4 Advantages and Disadvantages of ANN

3.4.1 Advantages of ANN

The basic advantages of ANNs are that they are tools that are data-driven, flexible and easy to handle non-linear processes. ANN has the ability to determine which model input is critical, so that there is no need for prior knowledge about the relationships among the variables being modelled. When developing neural network models, it is not necessary to know much knowledge about watershed characteristics and hydrological processes; and it is not needed to know the statistical characteristics of the input and output data. Furthermore, the non-stationarities in the data, such as trends and seasonal variations, are implicitly accounted for by the internal structure of the ANNs (Maier and Dandy, 1996). Due to the flexibility of ANN, different kinds of data can be included in an ANN model, both qualitative and quantitative (Xu and Li, 2002). Owing to the fact that ANNs do not need firm understanding of hydrological theory and watershed characteristics, the model calibration is relatively simple compared with other kinds of lumped hydrological models (Xu and Li, 2002). ANNs are highly parallel systems and contain many identical and independent components that can be executed simultaneously, often making them faster than alternative methods (Zealand et al., 1999). When an ANN has suitable architecture, it will be relatively insensitive to noisy data, unlike ARMA-type models. Therefore, an ANN model has the ability to determine the underlying relationship between model inputs and outputs, resulting in good generalization capability (Zealand et al., 1999).

3.4.2 Disadvantages of ANN

The black-box nature of ANNs and, therefore, inability to explain the reasoning for a result in a useful way is their single most significant weakness. The optimal network architecture as well as the optimal internal network parameters are somewhat problem dependent. The need for selecting a number of neural network parameters based on trial-and-error or rule-of-thumb characterizes ANNs' other inherent weakness (Tingsanchali and Gautam, 2000). For BPNNs, they require lots of supervised training with lots of input-output pairs. Additionally, the current training algorithm does not guarantee that the network will converge to a global optimization. Because the learning algorithms are based on gradient decent, the training processes are easily trapped into a local minimum. Finally, BPNNs can not cope with major changes in the system because they are trained on a historical dataset and it is assumed that the relationship learned will be applicable in the future (Zealand et al., 1999).

3.5 Important Issues Related to the Application of ANN

3.5.1 Generalization vs. Number of Hidden Nodes

In ANN applications, the number of input and output nodes is completely determined by the research problem itself while the number of hidden nodes is arbitrary. In general, more hidden nodes will provide an ANN greater capacity to capture the underlying relationships in training data. Imrie et al. (2000) indicated that too many hidden nodes will encourage each hidden node to memorize fluctuations in the training data that are not representative of the system being simulated, and thereby diminishing the generalization capabilities of the network. Zealand et al. (1999) showed that selecting too many hidden nodes will increase the training time but without significant improvement on training results. Generally, the best generalization ability is achieved when the fewest nodes are used.

The selection of a suitable number of hidden nodes is a tradeoff between an ANN model error and its generalization ability. The selection process is more an art than a science (Tingshanchali and Gautam, 2000). But Fausett (1994) still suggested a systematic way for the selection of hidden nodes. The general practice is to determine the number of hidden nodes by trial-and-error based on a total error criterion (ASCE Task Committee on Application of Artificial Neural Networks in

Hydrology, 2000 I).

3.5.2 Generalization vs. Training Time

For an ANN model, the longer it is trained, the smaller the error on training data could be achieved. ANN training should maximize its generalization ability rather than minimize the error on training dataset. So it is not necessarily to continue training until the neural network's error on training dataset reaches minimum. A common technique used to determine the appropriate training time is the early stopping technique. The technique uses two datasets, training and monitoring datasets, during training. The training algorithm adjusts the weights and biases according to training dataset. In certain interval during the training, the weights and biases are applied to calculate the simulation error on the monitoring dataset. As the training proceeds, the error on the training dataset will decrease constantly while the error on monitoring dataset will decrease in the initial phase then increases at a certain point. After this point, the network is starting to memorize the training pattern and starts to lose its ability for generalization. At this point, the early stop technique will stop the training. The early stopping technique was successfully implemented into the MATLAB Neural Network Toolbox. Recent researchers (Coulibaly et al., 2000; Imrie et al., 2000; Lin and Chen, 2004; Xu and Li, 2002) showed that early stop technique is a more systematic method to avoid overfitting, or to get maximum generalization ability.

3.5.3 ANN Input Selection

In the application of ANN, the selection of training data is important for the successful application of the ANN model (Xu and Li, 2002). Although an ANN is a data driven black-box model and the physical processes inherent in a research problem are often ignored, the physical processes are still important for the selection of ANN inputs. Through the learning process, ANN recognizes the relationship, both direct and indirect, between the inputs and the desired output values. It is obvious that the closer the relationships between inputs and outputs, the easier the training. So the relationship between inputs and desired outputs is the first priority in choosing the input variables. The firm understanding of physical processes, which determine how a set of input information is transformed into system output (quantity and timing of streamflow), will also help to choose input variables that have the most closest relationship to outputs.

The second issue associated with ANN inputs is providing ANN with as many patterns as possible. Once a network has been trained, the general relationship between the inputs and outputs contained in the training dataset has been obtained by the ANN architecture, the weights and biases. The network will generalize for new input data, which are within the scope of training dataset. On the other hand, the network will not necessarily generalize properly for the patterns located out of the range of the training dataset. To improve network performance, it is important that the training dataset provides a full and accurate representation of the problem domain (Zealand et al., 1999). For real-time streamflow forecasting, therefore, it is strongly advisable to include more data in the training dataset; in particular the peak flows with extreme large values. Even so, it is still advisable to retrain the model once a new dataset becomes available (Xu and Li, 2002).

The methodology of selection of ANN input involves autocorrelation analysis, cross-correlation analysis, physical considerations, and trial-and-error process. A sensitivity analysis can be used to determine the relative importance of a variable (Maier and Dandy, 1996).

3.5.4 Input Data Preprocessing (Scaling)

An ANN is theoretically able to handle raw data. However, without properly transforming the input data, the training process and the training result may not be the most efficient one. If variables in the training dataset vary greatly in magnitude, the weights and biases have to adapt to the difference in magnitude. The resulting weights and biases are not necessarily reflecting the importance of the input variables and will make the training algorithm runs inefficiently. Input data preprocessing/scaling is an efficient way to solve the problem. The commonly used scaling methods include: (i) linear transformation, (ii) statistical standardization, and (iii) mathematical functions. The linear transformation scales input into certain range between 0 and 1 or even 0.15

to 0.85. Scaling inputs down to range between 0.15 to 0.85 rather than 0 to 1 will take advantage of the most dynamic range of activation function and will give an ANN certain extrapolating ability. Nayak et al. (2004) and Sudheer et al. (2003) indicated that transferring input variable to normally distributed time series by the Wilson-Hilferty method will improve overall ANN performance. Smith and Eli (1995) transferred a hydrograph into Fourier series with 21 coefficients that were simulated by an ANN model. The authors found that the prediction of the entire hydrograph to be very accurate for multiple storm events. Imrie et al. (2000) showed that a cubic polynomial activation function in output layer will increase ANN extrapolation ability.

CHAPTER 4 Case Study and Results

4.1 Case Study Watershed, Data and Model Evaluation Criteria

4.1.1 Case Study Watershed --Cheakamus River above Daisy Lake Dam

To explore the applicability of ANN to flood forecasting, the Cheakamus River above the Daisy Lake Dam was selected as a case study watershed. The catchment is a coastal watershed located in southwest British Columbia, Canada. The drainage area of the Cheakamus River is about 945km^2, while the drainage area above Daisy Lake Dam is 721km^2. The glacier in the area is about 68km^2 or 9% of the case study area. The drainage basin of the Cheakamus River catchment and its general topographic features are shown in Figure 4.1.

The length of the catchment is about 12km along NNE-SSW and 36km along NE-SE. The elevation of the catchment varies from 379m to 2739m with a median elevation of 1375m (Figure 4.2). The northwest of the catchment is higher than the other parts of the watershed.

The Cheakamus River originates from Cheakamus Lake and flows to the Squamish River at the confluence near Brackendale, and then the Squamish River flows into Howe Sound, which is part of the Pacific Ocean.

The Cheakamus Project (Daisy Lake Dam) is a small hydropower plant built in 1957 and is operated by BCH. The Daisy Lake reservoir is approximately 6km long and 4300 hectares with storage capacity of 46,000,000m^3. The storage within the normal reservoir operation range of elevation 364.9m to 377.27m is about 40,000,000m^3. Power releases are discharged into the Squamish River upstream of its confluence with the Cheakamus River. Non-power releases from Daisy Lake Dam are discharged into the Cheakamus River.

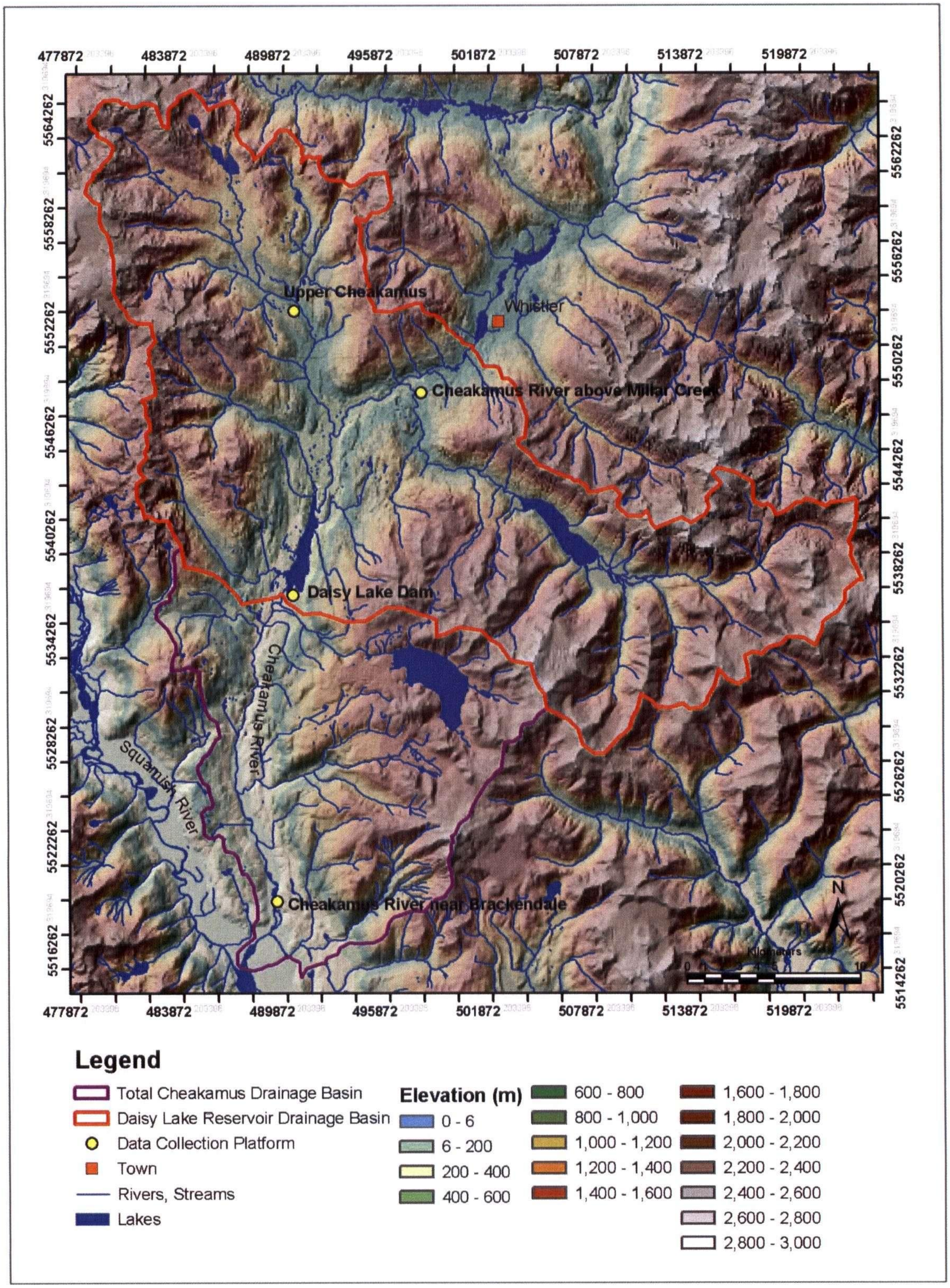

Figure 4.1 Cheakamus River catchment (BC, Canada)

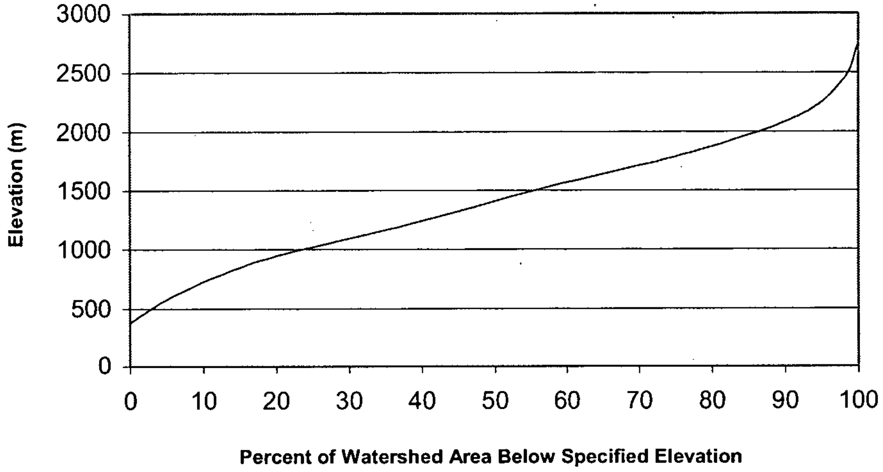

Figure 4.2 Hypsometric curve of Cheakamus catchment

Climate in the Cheakamus basin is dominated by the Pacific air masses, and is characterized by hot and dry summer and rainy winter. The confrontation of the Pacific air mass with the west-facing mountain slopes results in extremely prolonged rainfall in winter. The average annual total precipitation in Cheakamus watershed is 1944 mm (Upper Cheakamus DCP, 1986-2001). The precipitation in winter (October to January) is about 55% of the total annual precipitation; while it is only 15% in the period from June to September. In terms of average monthly precipitation, November is the highest (335mm) and July is the lowest (68mm) as shown in Figure 4.3. Snowpack in Cheakamus corresponds to its precipitation and temperature characteristics. In general, snow begins to accumulate in December and reach maximum in April. The snowmelt processes generally last for three months in the period of May to July.

The temperature in Cheakamus varies with seasons. The daily average temperature varies between –23℃ and 35℃. January is the coldest month with monthly average temperature of –2.3℃ while August is the hottest month with average temperature of 15.1℃. Figure 4.4 shows the mean, maximum and minimum daily temperature in each month at the Upper Cheakamus DCP.

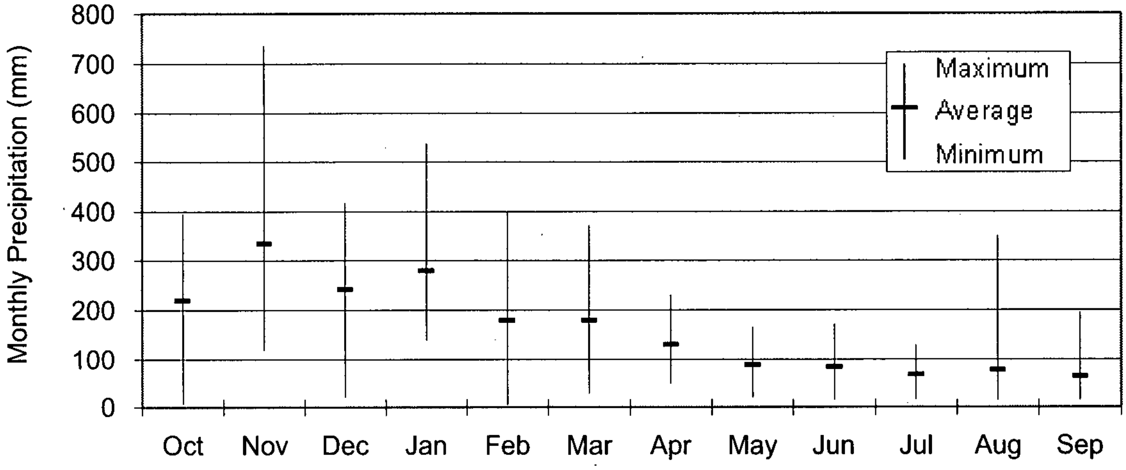

Figure 4.3 Monthly precipitation statistics at upper Cheakamus DCP (1986-2001)

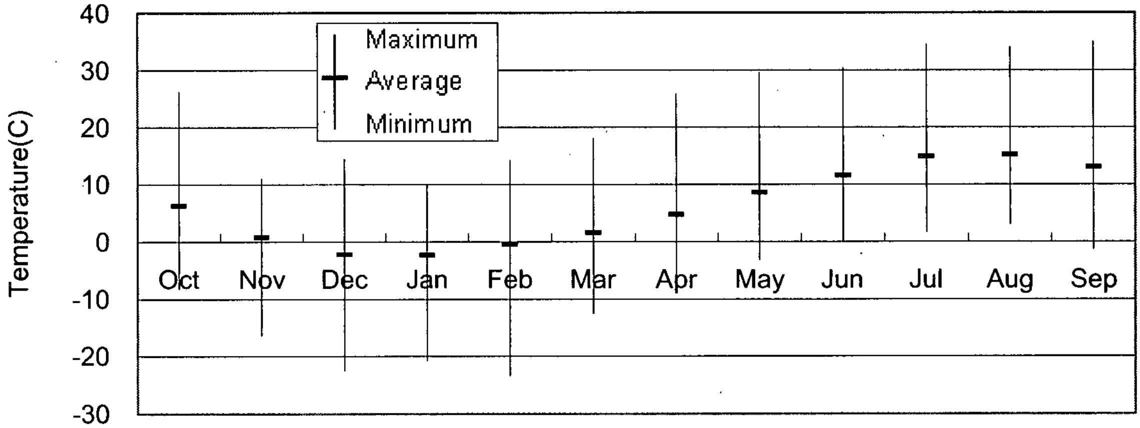

Figure 4.4 Daily temperature statistics at Upper Cheakamus DCP (1986-2001)

The average annual inflow to the Daisy Lake Reservoir is 49m^3/s. The monthly average inflows vary within the range of 20 to 115m^3/s, and are mainly affected by snowmelt. The monthly inflows from May to August vary from 68 to 115 m^3/s while the other months' vary from 20 to 43m^3/s. Figure 4.5 shows the similarity of monthly average inflow and temperature. Consistent with temperature, the lowest mean daily inflow occurs in January (20m^3/s) and the highest mean daily inflow occurs in June (115m^3/s).

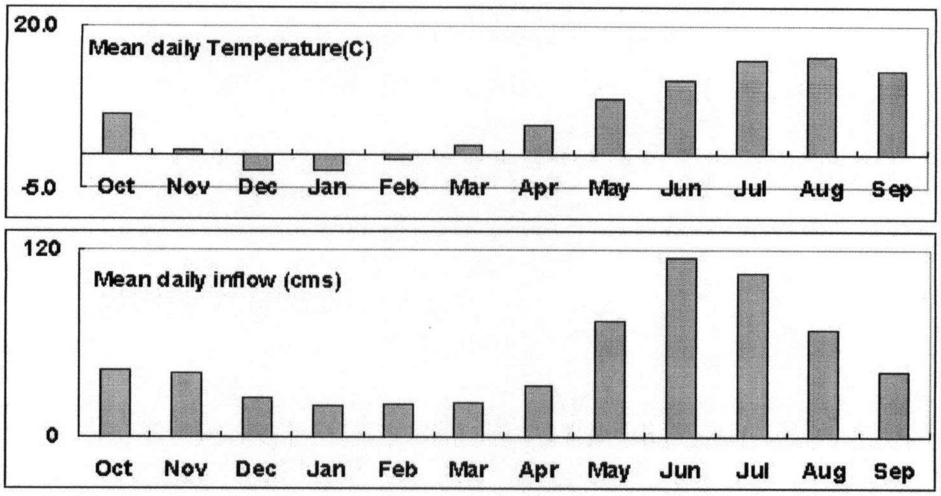

Figure 4.5 Comparison between monthly average inflow and temperature

Floods may occur in Cheakamus watershed at any time of a year (Figure 4.6), but floods with high magnitude mainly occur in the early winter season. There are three types of floods in the Cheakamus watershed: rainfall floods, snowmelt floods and rain-on-snow floods. The snowmelt runoff builds up the base flow and are similar to sinusoidal curve with one peak in each day. The snowmelt floods increase and decrease in a flat and fixed pattern, while the rainfall floods and rain-on-snow floods are flashy. It is relatively difficult to identify the rain-on-snow floods from the rainfall floods. The rain-on-snow floods are usually unexpectedly big in magnitude. The record highest flood occurred on November 11th, 1990, with a daily inflow of 647.8m^3/s and a return period of about 50 years.

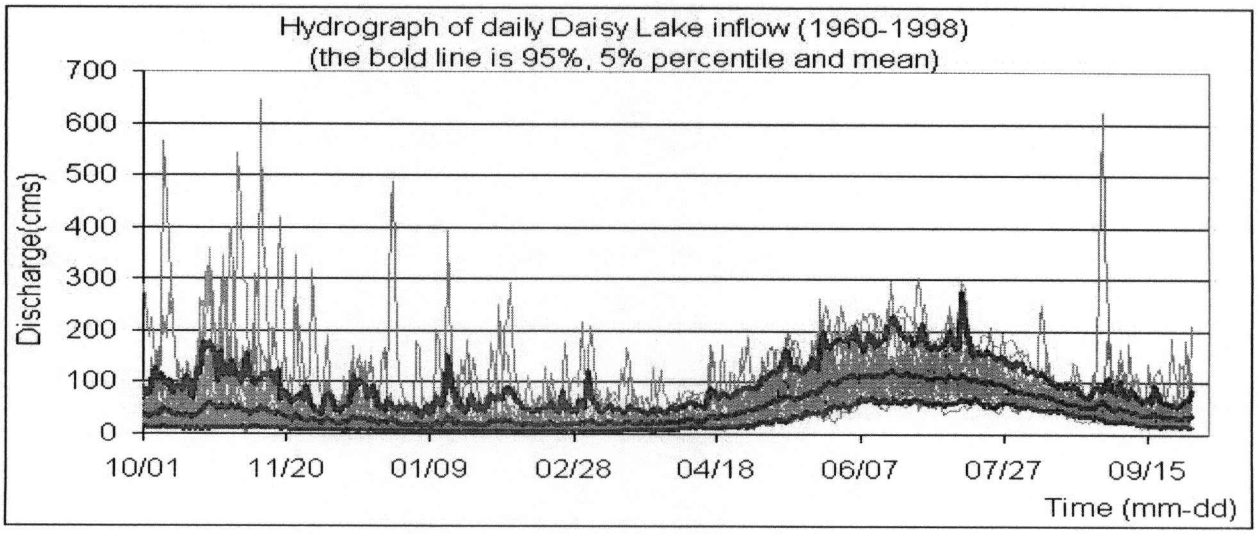

Figure 4.6 Historical daily inflow of Daisy Lake Reservoir

4.1.2 Data for Case Study

Historical daily precipitation, daily streamflow discharge and daily average temperature were obtained from BCH. The daily data cover the period from October 1, 1986 to September 1998 corresponding water year (October to September) 1987 to 1998. The daily data in the period of 1987 to 1995 are BCH well quality controlled official data that have been used in BCH RFS and the Water Use Planning process. The data in water year 1997 and 1998 are less quality controlled data. The daily precipitation and temperature data are observations of the Upper Cheakamus DCP (CMU), while the Daisy Lake inflow data were calculated by the reservoir mass balance equation.

Historical hourly precipitation, temperature, and Daisy Lake inflow used to train and test the hourly time step models were also obtained from BCH. The hourly data cover the period from October 1986 to September 1995. The hourly data are raw data and have not been quality controlled. The hourly precipitation and temperature are observations at the Upper Cheakamus DCP (CMU), the Daisy Lake inflows were calculated by the reservoir mass balance. The quality control of the hourly data, especially the inflow data was done by applying a moving average method and manual modification.

4.1.3 Model Evaluation Criteria

The evaluation and inter-comparison of different models or different sets of parameter are made by statistical indices comparison and hydrograph visual check. The statistical indices used in this research include the Root Mean Squared Error (*RMSE*), the Nash-Sutcliffe Coefficient of Model Efficiency (*CE*), the Mean Absolute Error (*MAE*), the volume error (*Vol.%*), the coefficient of correlation (*R*), and the parameters of linear regression between observed and simulated streamflows (*m* and *b*). The values of the indices are calculated according to Equations 4.1 to 4.5. The ideal values of the evaluation indices are shown in Table 4.1.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}[Q_{sim}(t) - Q_{obs}(t)]^2} \qquad (4.1)$$

$$CE = 1 - \frac{\sum_{t=1}^{n}[Q_{sim}(t) - Q_{obs}(t)]^2}{\sum_{t=1}^{n}[Q_{obs}(t) - \overline{Q}_{obs}]^2} \tag{4.2}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|Q_{sim}(t) - Q_{obs}(t)| \tag{4.3}$$

$$Vol.\% = \frac{\sum_{t=1}^{n}Q_{sim}(t) - \sum_{t=1}^{n}Q_{obs}(t)}{\sum_{t=1}^{n}Q_{obs}(t)} \times 100 \tag{4.4}$$

$$Q_{sim}(t) = m \cdot Q_{obs}(t) + b \tag{4.5}$$

where Q_{sim} is simulated streamflow (m^3/s); Q_{obs} is observed streamflow (m^3/s) and $\overline{Q}_{obs}$ is mean of the observed streamflow (m^3/s).

Table 4.1 Best value for model evaluation criterion

Statistical Indices	Abbreviation	Ideal Value
Root mean squared error	*RMSE*	0.0
Nash-Sutcliffe coefficient of model efficiency	*CE*	1.0
Mean absolute error	*MAE*	0.0
Volume error	*Vol.%*	0.0%
Coefficient of correlation	*R*	1.0
Linear regression parameter	*m*	1.0
Linear regression parameter	*b*	0.0

The CE is widely used to evaluate hydrologic model performance. It evaluates how close the simulated hydrograph matches the observed hydrograph. The CE varies from negative infinity to 1.0 corresponding to the worst and best agreement of hydrographs. The negative values of CE indicate that the observed mean is a better predictor than the model. Among these statistics, the RMSE and CE are usually the first two choices.

4.2 Daily ANN Model

4.2.1 Introduction

This case study intends to grasp the general knowledge of the ANN technique, explore the detailed application procedures, and find the factors that affect the runoff in the study watershed. The research intends to find if the ANN technique could be applicable in BC's small watersheds, especially those watersheds that are affected by snowmelt. In this case study, the ANN technique is applied to simulate the daily Daisy Lake inflow and compared with the UBCWM and a simple model ($Q_{t+1}=Q_t$) labeled as One Step Lag Model (OSLM), because both quality controlled daily data and the UBCWM can be easily obtained from BCH.

The application of the ANN model to simulate the daily streamflow involves four major steps: the ANN inputs selection, the input information preprocessing, the ANN architecture selection and model training. The ANN model used in this research is the Feedforward Error Backpropagation Neural Network with one hidden layer. The only parameter of the ANN architecture that needs to be determined is the number of hidden nodes.

Data in water year 1987 to 1990 were used to validate the ANN model. Data in water years 1991 to 1995 were used for model training/calibration, while the data in water years 1997 and 1998 were used for the ANN model performance monitoring during the training process as the early stop ANN training technique was used.

4.2.2 ANN Inputs Selection

The power of an ANN model stems from its ability of mapping input patterns to output patterns provided that the relationships between them do exist. It is obvious that the actual relationships between inputs and outputs are very important for an ANN model to perform properly. The relationship may be direct or indirect, linear or non-linear, single-to-single or multiple-to-single form. There are no mathematically rigorous ways to select appropriate input variables. The knowledge on hydrological processes, cross-correlation analysis, auto-correlation analysis and sensitivity analysis are generally employed to select an ANN model's inputs that could best represent the streamflow. The initial ANN model input variables were chosen according to the cross-correlation analysis between daily streamflow and precipitation (or temperature), and the autocorrelation (ACF) analysis of streamflow itself. These analysis techniques will indicate the direct relationship between streamflow and precipitation (or temperature, past streamflow). Figure 4.7 shows the ACF of streamflow (*Q*). The ACF decreases with the increase of the lag days. The correlation between streamflows in two successive days, Q_t and Q_{t-1}, is as high as 0.93. Figure 4.7 clearly indicates that past streamflows are strong indicator of future streamflows. The autocorrelation analysis just shows the single-to-single relationship and could only suggest past streamflows are good indicators. It is difficult, however, to determine how many past streamflow information is good due to the complexity of multiple-to-single relationship. The complexity is caused mainly by the combination and the joint distribution of past streamflows. In this research, streamflows of past two days, Q_t and Q_{t-1}, were chosen as the initial past streamflow information to forecast the streamflow at $t+1$. The two streamflows are highly correlated to future streamflow and could inform the ANN model about the streamflow trends.

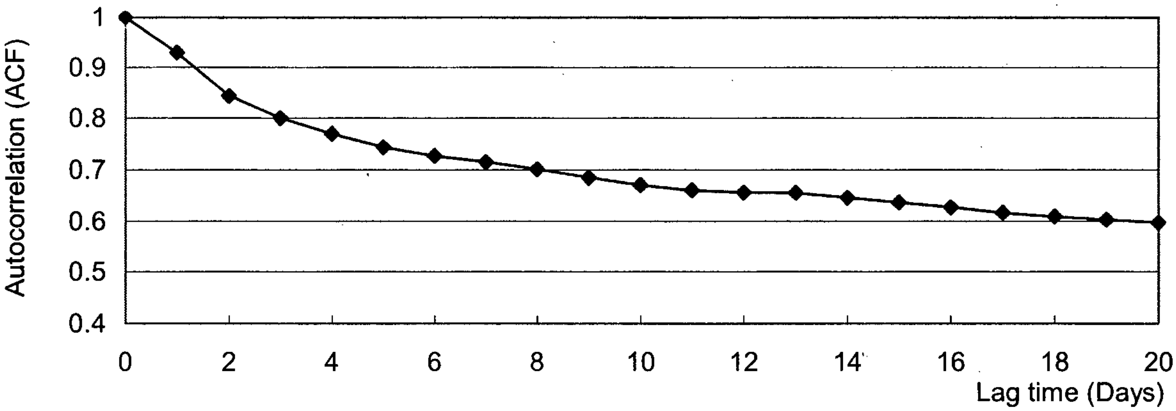

Figure 4.7 Streamflow autocorrelation (ACF) analysis

Figure 4.8 shows the cross-correlation between streamflow and other information such as precipitation, maximum daily temperature and minimum daily temperature. Figure 4.8 indicates that precipitation at lag 0, 1 and 2 days, which correspond to p_{t+1}, p_t and p_{t-1}, have positive correlation with streamflow Q_t. Hence, for day $t+1$ the daily precipitations, p_{t+1}, p_t and p_{t-1}, are thought as the initial precipitation indicators. The small value of cross-correlation between streamflow and precipitation does not necessarily mean that precipitation is not important, it is caused by the large number of zero precipitation data. The cross-correlation between streamflow and precipitation is relatively high in flood period (refer to sub-section 4.3.2).

The temperature affects streamflow indirectly by changing the form of precipitation (e.g. rainfall or snowfall) and melting the snowpack, which consequently changes the watershed water storage (*S*). The rainfall–flood response time is about 6~10hrs in the case study watershed, and accordingly the value of the temperature-flood response time should be longer. When considering that the streamflow indicators, Q_t and Q_{t-1}, already contain the effects of past temperature, the temperature indicators selected are only future daily maximum temperature ($Tmax_{t+1}$) and daily minimum temperature ($Tmin_{t+1}$).

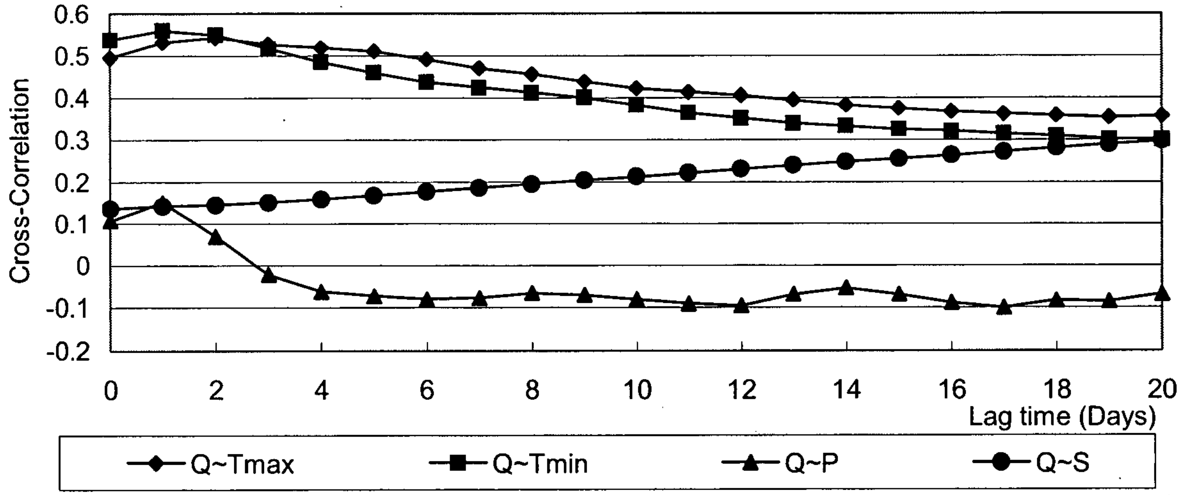

Figure 4.8 Daisy Lake inflow cross-correlation analysis

The case study watershed is heavily affected by snowmelt runoff especially in the summer. Previous applications of ANN used the snow water equivalent (SWE) or snowpack information as an indicator (Zealand et al., 1999; Coulibaly et al., 2000). Due to the low accuracy and representative of the snowpack data, it was not chosen as indicator in the research. Because of the availability of a model for SWE calculation in the case study watershed, and the high uncertainty associated with the SWE calculation results, SWE was not chosen as indicator in the research too, while a new indicator, water storage trend (S), was introduced in this research. S is calculated by:

$$S(t+1) = S(t) + p(t+1) - q(t+1) - E \qquad (4.6)$$

Where p is the daily precipitation (mm), q is the daily streamflow volume in terms of runoff height (mm) and E is the daily evaporation (mm).

In summary, the initial input indicators selected for one day ahead ($t+1$) streamflow forecasting are the future precipitation (p_{t+1}), the future temperatures ($Tmax_{t+1}$, $Tmin_{t+1}$), the past precipitations (p_t, p_{t-1}), the past streamflows (Q_t, Q_{t-1}) and the past water storage (S_t). The initial inputs selected were further refined by sensitivity analysis which was focused on the number of past precipitation and streamflow data points.

4.2.3 ANN Inputs Preprocessing

The preprocessing of ANN inputs includes information encoding and scaling. The encoding is used for non-digital and non-continuous information, therefore it has nothing to do with this research. This research used the simplest linear transformation to scale input information down to 0.15 to 0.85, which is considered to be the most dynamic range of the sigmoidal activation function. The equation used for scaling is:

$$v_{new} = 0.15 + \frac{v_{old} - v_{\min}}{v_{\max} - v_{\min}} \times 0.7 \qquad (4.7)$$

Where v_{old} is the original hydrometric data value, v_{min} is the minimum of the original input time series and v_{max} is the maximum of the original input time series.

The input preprocessing methods, such as logarithm transformation and Wilson-Hilferty transformation, were not adopted, as these transformations will dramatically change the statistical properties of the input and the output data, and will distort the physical relationship inherited in rainfall-runoff processes. For example, the logarithmic transformation will mask the most important high flows and expose the low flow data.

4.2.4 ANN Training

The ANN model training is a process of adjusting the ANN weights. The training process tries to find a set of parameters (weights) that makes the ANN simulated streamflows match observed streamflows as close as possible. In an ANN application, the training is also a process of determining the ANN architecture in terms of number of hidden nodes. The number of hidden nodes is generally found by training the ANN with different number of hidden nodes, then choosing the ANN with the best average performance. In this research, ANN training was also used to find the best combination of input indicators by means of sensitivity analysis. Due to the large number of weights and the gradient descent training algorithm, training could be easily trapped by a local minimum. To avoid this problem, training was repeated many times with different initial weights. Each training gives a set of parameter, then the set of parameters which best simulate validation dataset was chosen as the final set of parameters.

In this research, the training data were divided into three parts namely training dataset (1991-1995), monitoring dataset (1997-1998) and validation dataset (1987-1990). The early stop technique was used in the ANN training.

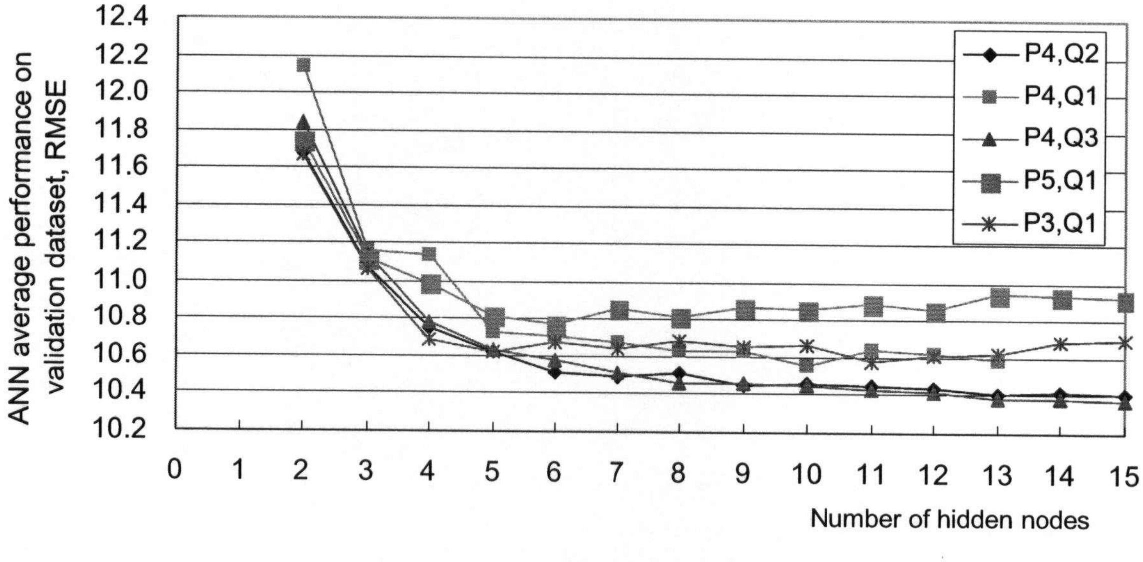

Figure 4.9 ANN input indicator sensitivity analysis

Figure 4.9 shows the ANN input indicators refinement process, or the ANN inputs sensitivity analysis. The sensitivity analysis was focused on choosing the number of past precipitations and streamflows. The *P* and *Q* in Figure 4.9 are precipitation and past streamflow, respectively. The number following *P* and *Q* are the number of precipitation and streamflow indicators used. The figure indicates that the ANN models using four precipitations and two or three streamflow records (P4Q2 and P4Q3) perform better than others. The difference between P4Q2 and P4Q3 is negligible while the ANN model using two past streamflow indicators is slightly better when the number of hidden nodes is less than seven. For the sake of simplicity and better generalization ability, two past streamflow indicators are thought to perform better. The P4Q2 ANN model for daily streamflow forecasting is shown in Equation 4.8 and Figure 4.10. The model has nine input indicators.

$$Q_{t+1} = f\left(p_{t+1}, p_t, p_{t-1}, p_{t-2}, Q_t, Q_{t-1}, T\max_{t+1}, T\min_{t+1}, S_t\right) \tag{4.8}$$

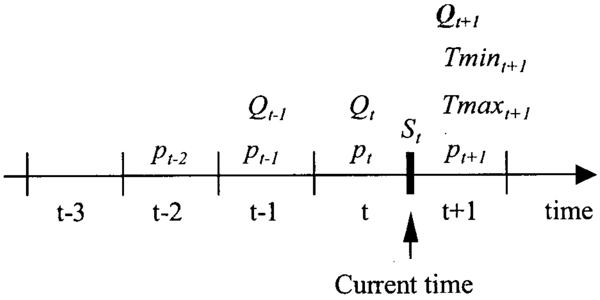

Figure 4.10 The input indicators to daily time step ANN model (ANN9-5-1)

Figure 4.11shows the trial-and-error process that is used to choose the number of hidden nodes. For a fixed combination of input indicators, those ANNs with different number of hidden nodes were trained and the ANN performance on validation dataset was calculated in term of RMSE. The best ANN performance is chosen from the results of many trainings to draw the curve. As shown in Figure 4.11, the P4Q2 ANN model has the best performance when the number of hidden nodes is five. The comparison among the models also indicates that P4Q2 model with five hidden nodes was the best one. The P4Q2 ANN model with five hidden nodes is denoted as ANN9-5-1 model, in which 9 represents nine input indicators, 5 represents five hidden nodes and 1 means that the ANN model has only one output node representing the forecasted streamflow.

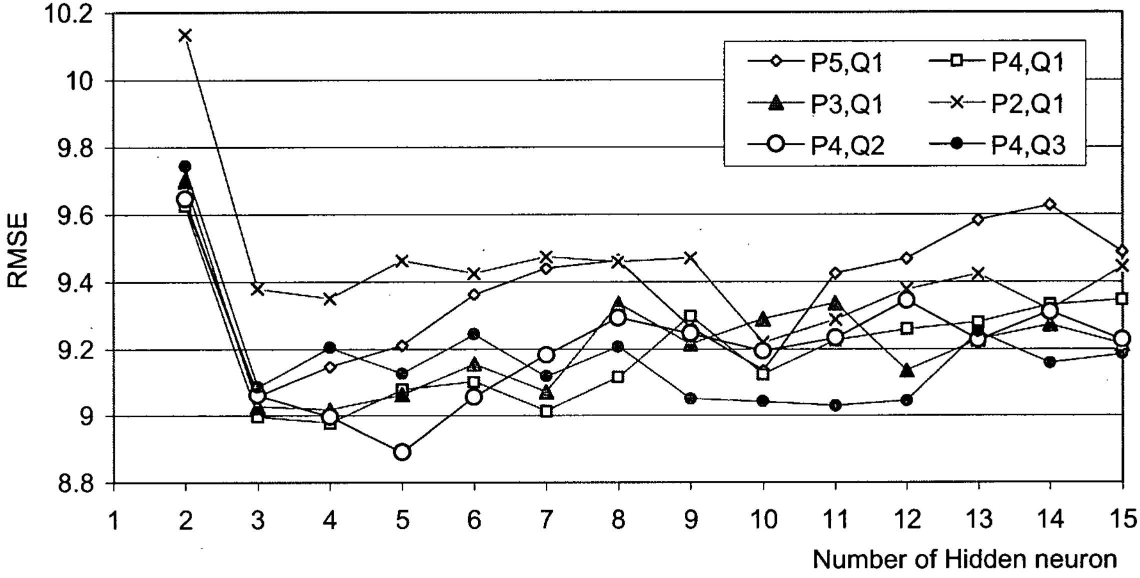

Figure 4.11 The number of hidden neurons selection for ANN9-5-1 model

4.2.5 ANN Model Training Results

Table 4.2 lists the ANN9-5-1 model training results for training, monitoring and validation datasets. The modelling results indicate that ANN9-5-1 model's overall performance in the watershed is good. In the model validation period (1987-1990), ANN9-5-1 model's streamflow simulation error, in terms of MAE, is 5.22 m^3/s or 11.1% of the average streamflow in the period. The low MAE reflects that the model has low average modelling error for various streamflow situations (e.g. high flows and low flows, or flows in different seasons). In the validation period, the model has a high CE of 0.95, which indicates the simulated and observed hydrograph matches each other very well; the model simulates the total runoff volume at a very low percent error of 0.3%. For summary, the ANN9-5-1 model's high performance on modelling streamflow value, shape of hydrograph and total volume of runoff indicates that the model is good for the research watershed.

The comparison of various statistic indices between model calibration and validation periods (Table 4.2) shows that ANN9-5-1 model performance is consistent in the two periods, and indicates that the model will have the same performance on future datasets. ANN9-5-1 model's

MAE statistics are 6.98 and 5.22m^3/s (Table 4.2) for calibration and validation periods, respectively. The MAEs are equivalent to 11.1% and 12.5% of average streamflow in the two periods, respectively. The model also has similar CEs, 0.93 and 0.95, in calibration and validation periods, respectively. Beside the MAE and CE statistics, the total runoff volume simulation error values, 0.3% and 1.2%, in the two periods are also close to each other.

The similar performance of the ANN9-5-1 model in calibration and validation periods indicates that the ANN training neither over fits nor under fit the calibration dataset. The over fitting will cause the model to perform bad on validation dataset, while the under fits will cause the model to perform on validation dataset in a uncertain way. This further indicates that the ANN training method, LMBP coupled with early stop technique being adopted in the research, is a practical and good method for ANN training.

Table 4.2 ANN9-5-1 model training results

YEAR	Duration	Mean of Obs. (cms)	RMSE (cms)	CE	m	b (cms)	r	MAE (cms)	Vol.Er. (%)
1987	1986/10/01-87/09/30	51.1	8.92	0.96	0.94	3.47	0.978	5.54	0.7
1988	1987/10/01-88/09/30	46.0	8.44	0.95	0.90	2.56	0.976	5.02	-4.3
1989	1988/10/01-89/09/30	43.4	7.80	0.95	0.94	3.31	0.976	4.8	1.7
1990	1989/10/01-90/09/30	47.3	9.80	0.92	0.94	4.19	0.962	5.54	3.0
1991	1990/10/01-91/09/30	66.0	14.3	0.96	1.00	2.75	0.98	8.44	3.9
1992	1991/10/01-92/09/30	56.6	12.6	0.92	0.92	5.3	0.959	7.1	1.1
1993	1992/10/01-93/09/30	45.8	9.6	0.95	0.94	4.16	0.975	5.36	3.4
1994	1993/10/01-94/09/30	53.5	12.0	0.92	0.92	5.23	0.962	6.8	2.0
1995	1994/10/01-95/09/30	56.3	10.2	0.96	0.93	3.96	0.979	5.86	0.5
1997	1996/10/01-97/09/30	61.5	14.4	0.91	0.86	5.45	0.959	8.31	-5.5
1998	1997/10/01-98/09/30	53.1	9.75	0.94	1.01	1.72	0.973	5.92	4.3
Calibration	1991-1995	55.6	11.9	0.95	0.96	3.62	0.973	6.71	2.2
Validation	1987-1990	46.9	8.85	0.95	0.93	3.35	0.973	5.22	0.3

Table 4.2 also shows that the ANN9-5-1 model performance is consistent between years in the model calibration and validation period (1987-1995). The CEs for those years vary from 0.92 to 0.96. The MAEs for those years vary from 10.2% to 12.8% of corresponding average annual streamflow. The total runoff volume simulation errors vary in a small range of –4.3% to 3.9%. The consistent high performance of ANN9-5-1 model in nine years indicates that the model has

adapted to the complicated non-linear rainfall-runoff mechanism in the research watershed.

Figures 4.12 to 4.15 present the hydrograph comparison of observed streamflow and ANN9-5-1 model simulated streamflow for validation dataset (1987-1990). The figures show that simulated hydrographs match observed hydrographs very well. The good match of the two hydrographs explains the high CEs of each validation year in Table 4.2. The watershed is characterized by: (1) rainy winter and dry summer, (2) average low streamflow in winter and high streamflow in summer and (3) snowpack highly affects the streamflow. It is shown by Figures 4.12 to 4.15 that the ANN9-5-1 model simulates low flows (about 10-20m^3/s), medium flows (20-50m^3/s) and high flows (>50m^3/s) equally well. The model does not show obvious simulation bias for low, medium and high flows. This indicates that the ANN9-5-1 model has adapted to the runoff patterns which can be roughly classified into groundwater runoff, soil water runoff and snowmelt runoff. It is one of the signals that indicates the ANN9-5-1 model has adapted to the non-linear rainfall-runoff response in the watershed.

As discussed in section 3.2, the ANN architecture provides an ANN model with the power to simulate a system or represent any continuous function exactly (Fausett 1994, Charpter 6.1). The system/function can be linear or non-linear. The research in this section shows that a simple ANN model with three layer and nine nodes in hidden layer could handle the non-linear and complicated rainfall-runoff response in the case study watershed with an acceptable accuracy. The ANN training method (LMBP and early stop technique) used in the research makes the parameter set that resulted from training to have the biggest generalization ability for future data. In other words, this makes the performance of an ANN model on the training (calibration) dataset and validation dataset as equal as possible (Table 4.2).

The high performance of the ANN9-5-1 model also benefits most from one of its features, which dictates that an ANN model could use any kind of information as model input. As shown by Equation 4.8, the ANN9-5-1 model uses streamflow in past two days as model input. Obviously, the streamflow input will strongly show the model current watershed and streamflow conditions. The two streamflow inputs have the possibility to show the model the streamflow trend as well as streamflow magnitude. Taking past streamflow information into an ANN model provides the model higher starting modelling accuracy than conventional rainfall-runoff models, which do not use past streamflows, the most accurate and valuable information, as model input.

The watershed water storage trend (*S*) being introduced in ANN9-5-1 model takes the most advantages of available high quality data (precipitation, streamflow and evaporation) and informs the model about the relative water storage in the watershed. *S* has less uncertainty than snowpack information and SWE, which have been used in previous snowmelt runoff simulation (Zealand et al., 1999; Coulibaly et al., 2000). Using *S* in the model increases the model input information quality and facilitates the model getting higher modelling accuracy.

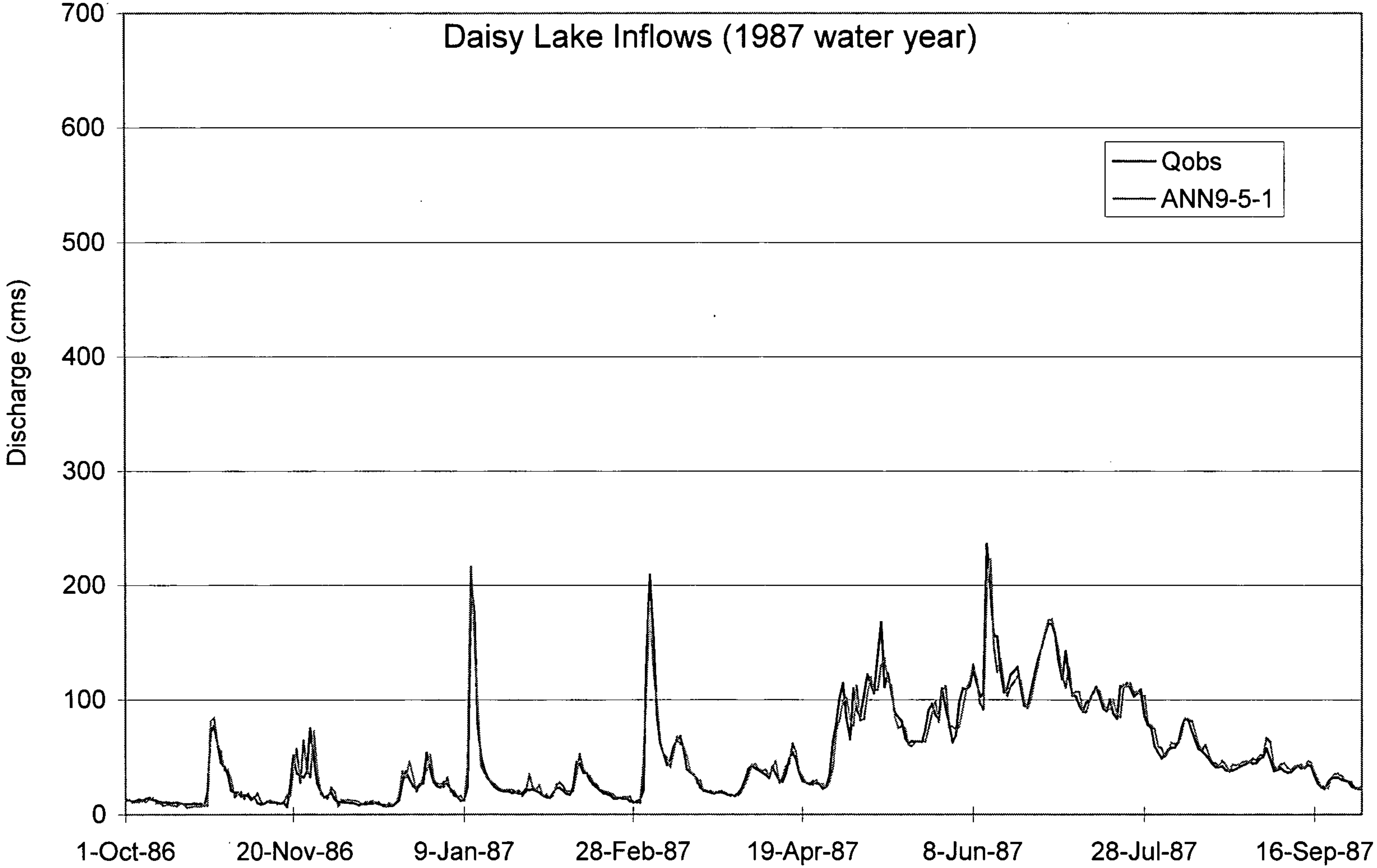

Figure 4.12 ANN9-5-1 model validation hydrograph (1987)

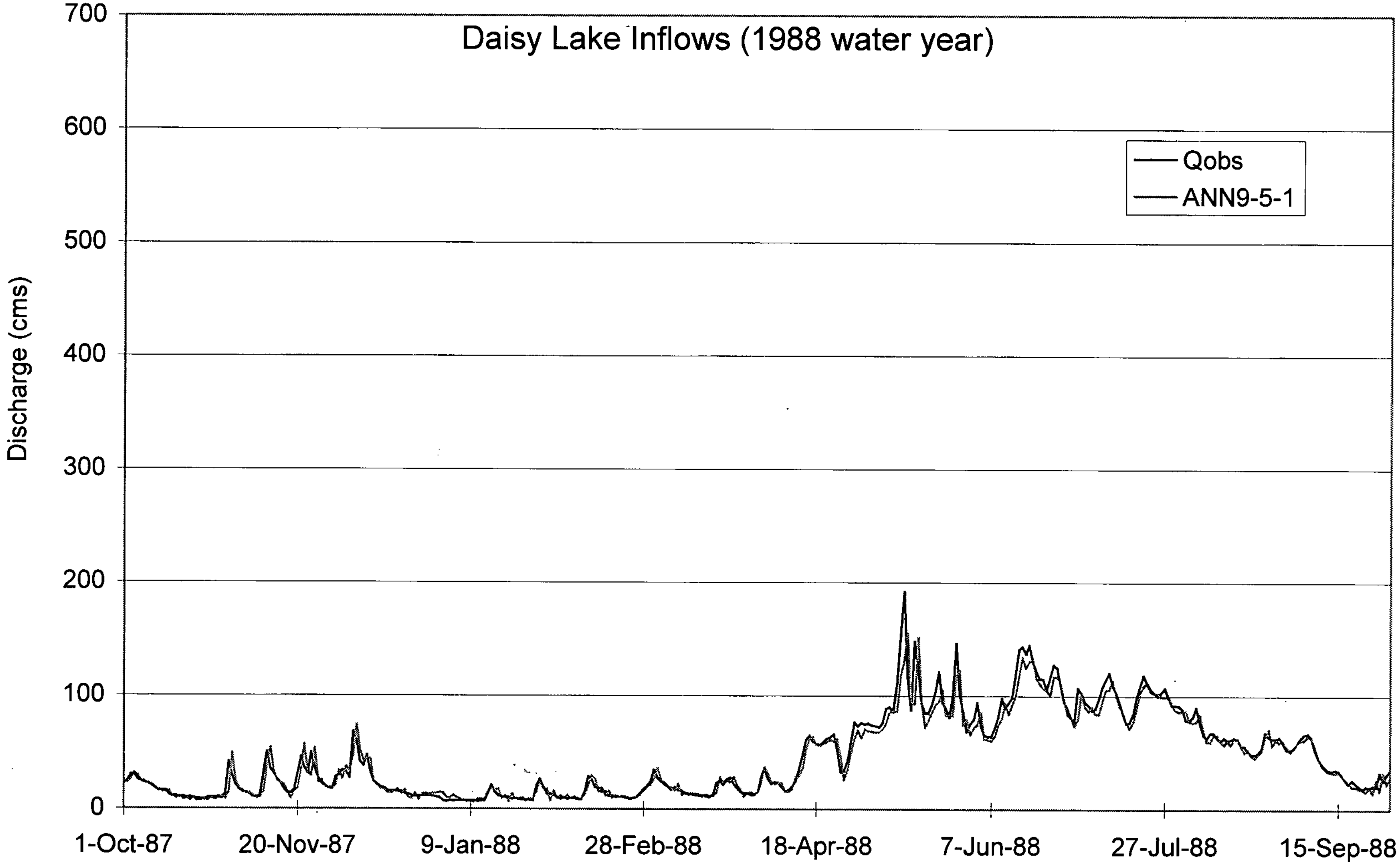

Figure 4.13 ANN9-5-1 model validation hydrograph (1988)

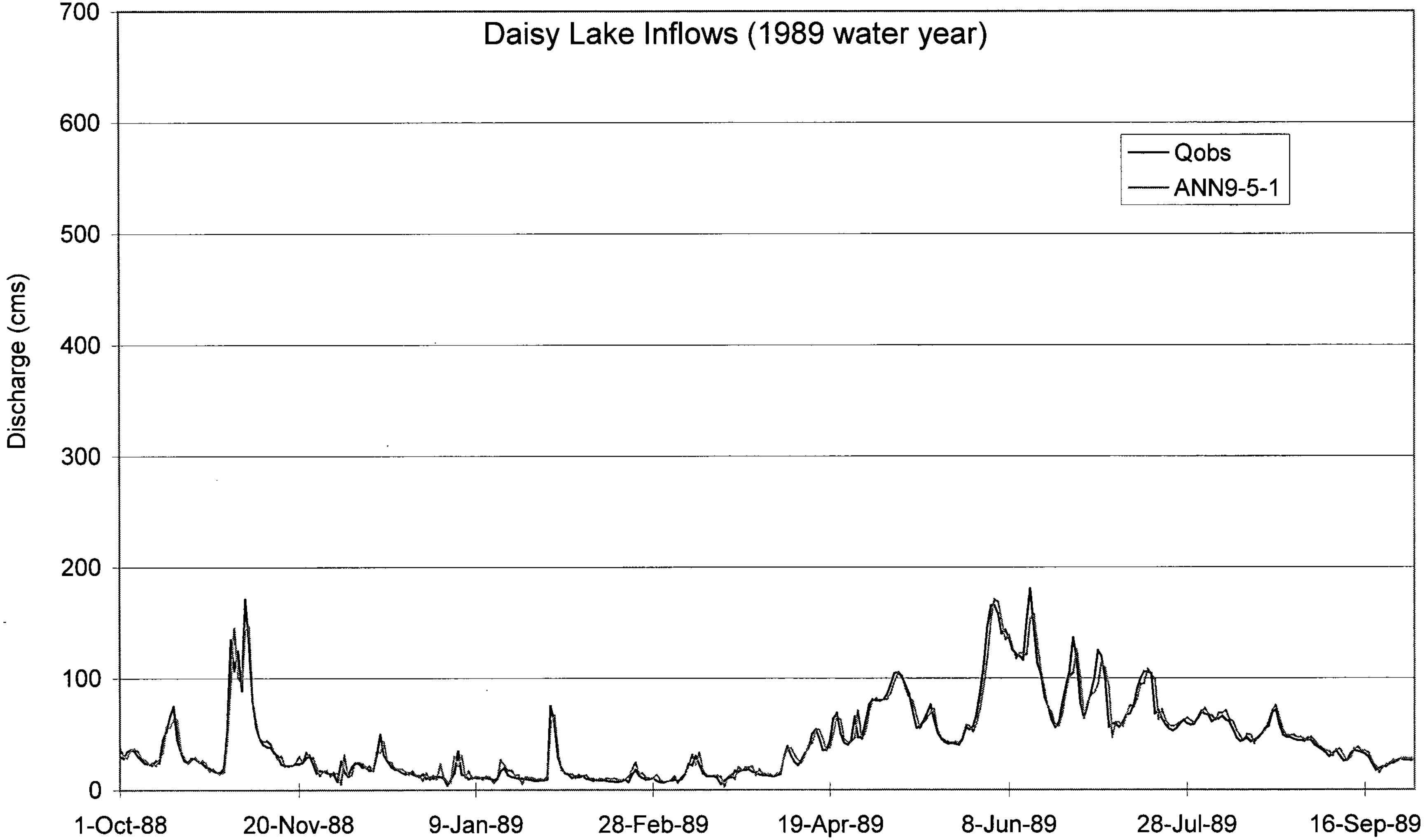

Figure 4.14 ANN9-5-1 model validation hydrograph (1989)

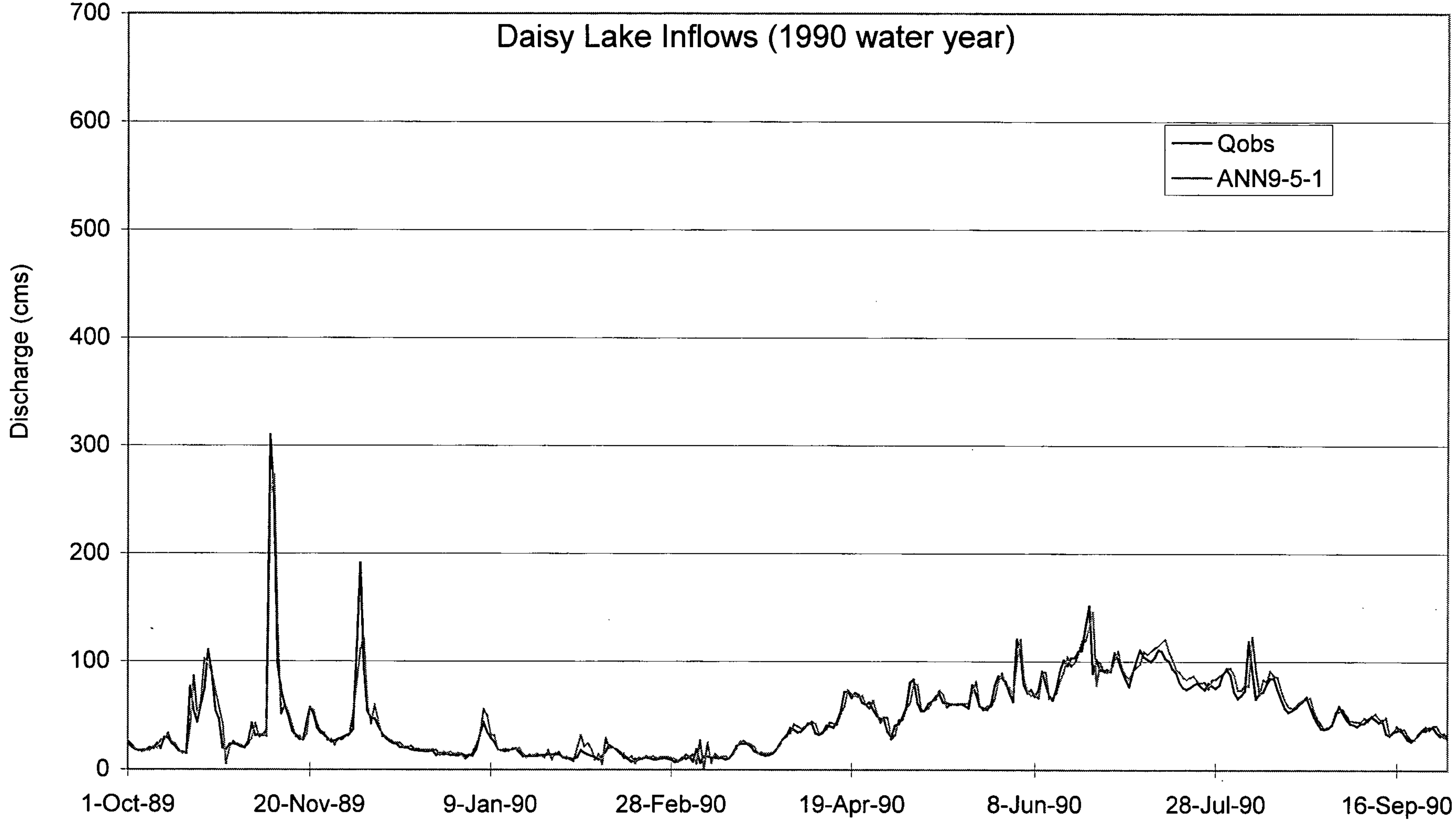

Figure 4.15 ANN9-5-1 model validation hydrograph (1990)

4.2.6 ANN, UBCWM and OSLM Performance Comparison

As shown in chapter 4.2.5, the ANN performance was good in the case study watershed. To judge the ANN model performance in a more detailed way, this research compared the ANN performance with that of other two reference models, UBCWM (Quick, 1995) and OSLM. The performance of the two reference models were calculated and compared with that of the ANN model as shown in Figures 4.16 to 4.18 and in Table 4.3. The UBCWM and ANN9-5-1 models used the same type of hydrometric and climatic information from same hydrometric station. The two models used data in the same period for calibration and testing. The OSLM does not involve any calibration and validation due to its model structure ($Q_{t+1}=Q_t$). Both the OSLM and ANN9-5-1 models used past streamflow as input. The ANN9-5-1 model performance improvement, compared to the OSLM, comes from the ANN technique and the additional indicators used. Figures 4.16 and 4.17 show the *RMSE* and *MAE* comparison among the three models. The bar chart indicates that the ANN modelling error is constantly less than that of UBCWM and OSLM in both training period (1991-1995) and the validation period (1987-1990).

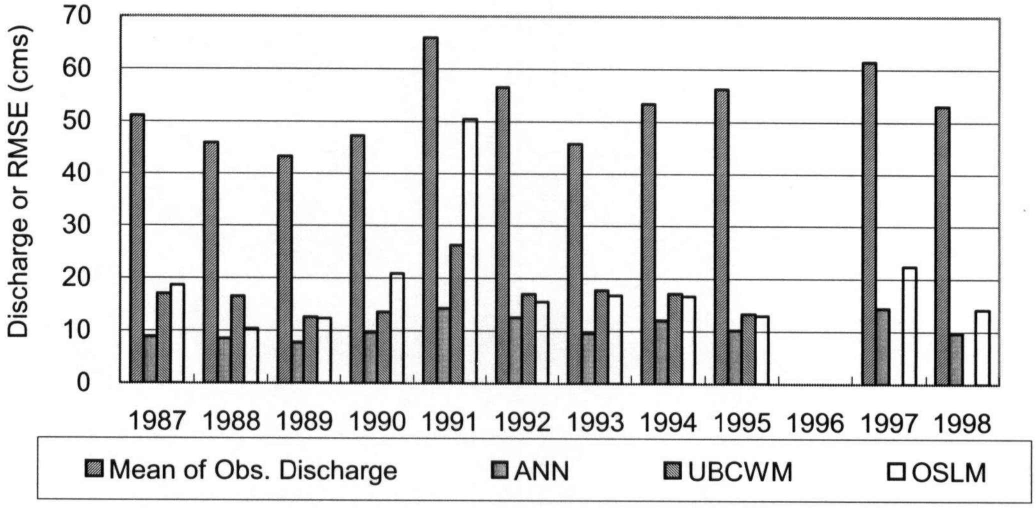

Figure 4.16 ANN, UBCWM and OSLM performance comparison (*RMSE*)

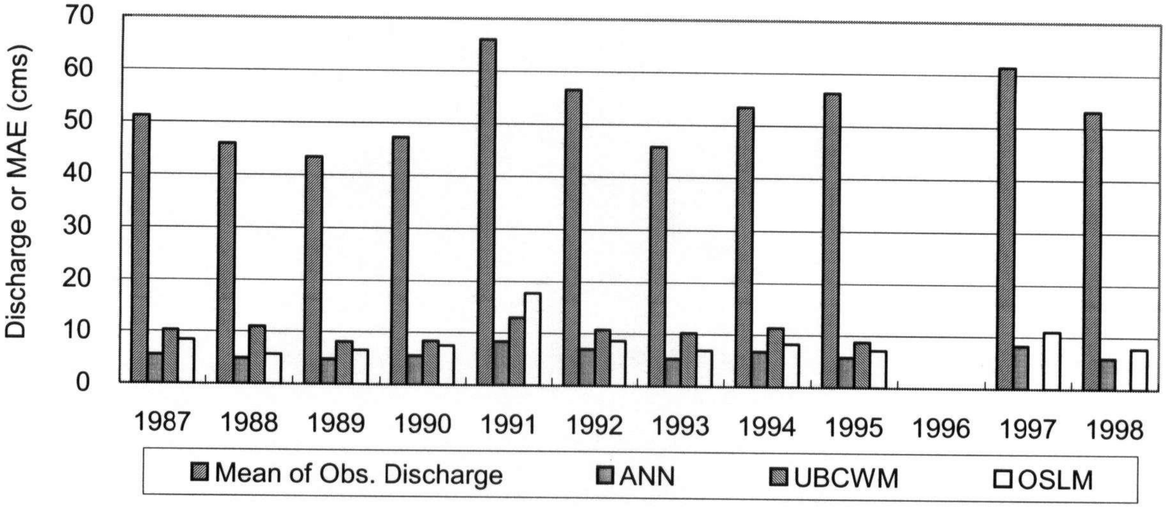

Figure 4.17 ANN, UBCWM and OSLM performance comparison (*MAE*)

Figure 4.18 shows that the ANN modelling efficiency *CE* is greater than that of the other two reference models. The *CEs* of the ANN model are greater than 0.9 in all the calibration and validation years, and the average *CE* in the training and the validation periods are all 0.95, which is greater than the average *CE* for the UBCWM in either the calibration (0.86) or the validation periods (0.84). The OSLM modelling efficiency is higher than that of UBCWM in six out of nine years. The bad modelling efficiency of the OSLM on 1990 and 1991 data is caused by the large number of floods in those two years. The comparison of *CE* between UBCWM and OSLM indicates that the UBCWM performs better than the OSLM in flooding periods. But in non-flood periods, the UBCWM performance is lower than the OSLM, while for the ANN model , its performance is consistent in both high and low flow periods.

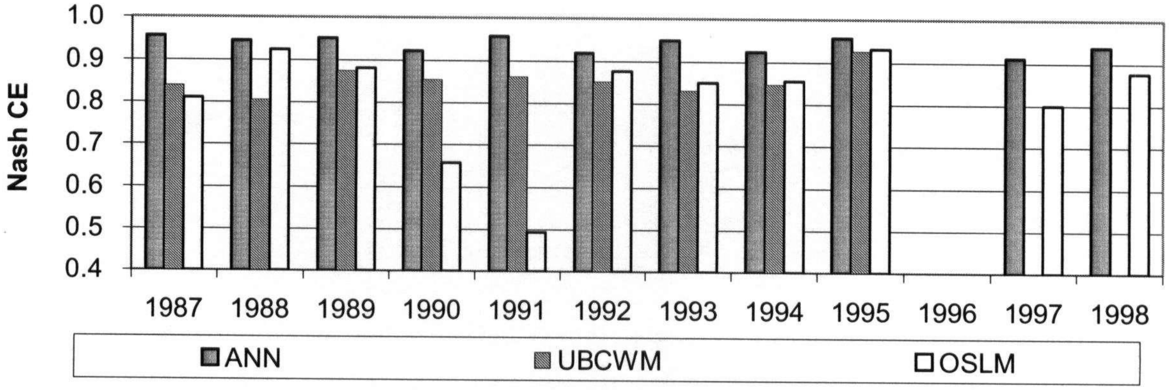

Figure 4.18 ANN, UBCWM and OSLM performance comparison (CE)

Table 4.3 Statistics comparison among ANN9-5-1, UBCWM and OSLM

YEAR	Duration	Mean of Obs.	RMSE			MAE			CE		
	(YY/MM)	(cms)	ANN	UBC	OSLM	ANN	UBC	OSLM	ANN	UBC	OSLM
1987	1986/10-87/09	51.12	8.92	17.08	18.63	5.54	10.32	8.38	0.96	0.84	0.81
1988	1987/10-88/09	45.95	8.44	16.52	10.29	5.02	11.07	5.75	0.95	0.81	0.92
1989	1988/10-89/09	43.38	7.80	12.60	12.35	4.80	8.17	6.64	0.95	0.88	0.88
1990	1989/10-90/09	47.28	9.77	13.57	20.83	5.54	8.41	7.55	0.92	0.85	0.66
1991	1990/10-91/09	66.01	14.29	26.34	50.56	8.44	12.99	17.71	0.96	0.86	0.49
1992	1991/10-92/09	56.61	12.55	17.02	15.59	7.10	10.85	8.66	0.92	0.85	0.88
1993	1992/10-93/09	45.78	9.63	17.77	16.81	5.36	10.34	6.93	0.95	0.83	0.85
1994	1993/10-94/09	53.45	12.03	17.15	16.70	6.80	11.35	8.36	0.92	0.85	0.86
1995	1994/10-95/09	56.32	10.21	13.32	12.97	5.86	8.87	7.21	0.96	0.93	0.93
1997	1996/10-97/09	61.54	14.43		22.30	8.31		11.09	0.91		0.80
1998	1997/10-98/09	53.08	9.75		14.21	5.92		7.84	0.94		0.88
Cal.	1991-1995	55.64	11.89	18.82	26.56	6.71	10.88	9.77	0.95	0.87	0.74
Val.	1987-1990	46.93	8.85	15.08	16.12	5.22	9.49	7.08	0.95	0.84	0.82

Note: UBC represents UBCWM

Figures 4.19 to 4.22 present the hydrograph comparisons of the ANN9-5-1 model and UBCWM for the validation dataset (1987-1990). Figures 4.23 to 4.26 are the same comparisons for the calibration dataset. As discussed in 4.2.5, the ANN9-5-1 model simulated hydrographs match observed hydrographs very well in various aspects. The UBCWM simulated low flows in autumn to spring are good, but it does not simulate the high flows in snowmelt season as good as low flows. The UBCWM simulated flows in snowmelt season deviate from observed flows in various patterns. In 1988, the UBCWM constantly underestimates streamflows; while in 1989 it overestimates streamflows. In 1987, it underestimates streamflows for about one month then overestimates streamflows for about two months. The bad performance of UBCWM in snowmelt season indicates that its model structure or its parameters are not properly set for snowmelt runoff modelling. Assuming that UBCWM's structure and parameters are good, its bad performance are still showing that the model is too sensitive to low quality data, or it does not have the ability to adapt to watershed status quickly, or it does not have enough feedback mechanism to keep the model runs on the right track.

The ANN9-5-1 model is not sensitive to noisy input information by nature and it does not show

UBCWM's drawbacks. Recall that a hidden node of an ANN model calculates the weighted sum of inputs to itself, then transfers the weighted sum by a sigmoidal function (Equation 3.1 or 3.2, and 3.5 or 3.6). The sigmoidal function (Figure 3.4 or Figure 4.5) is bounded and its output does not change so much when its input, which is the weighted sum of ANN model input, is changed dramatically. For example, the binary sigmoidal function's output will change from 0.9820 to 0.9997 when the weighted sum of ANN model inputs changes from 4 to 8. So for an ANN model, the noisy inputs will cause this weighted sum at a node to change in a big range first, but the node's output will be smoothed out by the sigmoidal function. The ANN architecture and algorithm will filter certain parts of noisy inputs, which make an ANN model insensitive to noisy input by nature. The ANN9-5-1 model is even more insensitive to low quality information (e.g. snowpack, snow water equivalent) by using high quality data (e.g. precipitation, previous streamflow, evaporation). The ANN model uses most recent watershed status information (e.g. streamflow, water storage trend) to forecast streamflow for the next time step. Compared with UBCWM, the ANN9-5-1 model has more accurate forecast starting point, it is more suitable for real-time streamflow forecasting.

In terms of flashy flood simulation, Figures 4.19 to 4.26 show that UBCWM's performance is better than its performance on snowmelt runoff simulation. However, when compared with that of ANN9-5-1 model, UBCWM's performance on flashy flood simulation is less successful than ANN9-5-1 model. UBCWM tends to underestimate major floods (e.g. the two flood events in February and March 1987; one flood event in November 1988; two flood events in November and December 1990; five major flood events in 1991, etc.). UBCWM simulated major flood events recession processes are slower than the observed ones (e.g. one flood event in November 1989; five flood events in 1991). The performance of the UBCWM indicates that it does not fully capture the runoff components and runoff routing processes. Some of the runoff components control the volume of a flood and the baseflow of flood peak while the runoff routing will affect the streamflow rising and recession speed and the magnitude of peak. For summary, UBCWM does not reflect some parts of the runoff mechanism in the case study watershed. The hydrograph comparison between UBCWM and ANN9-5-1 models indicates that ANN9-5-1 model reflects the runoff mechanism or the non-linear rainfall-runoff responses better than UBCWM in the watershed. The ANN9-5-1 model is more dynamic than UBCWM in streamflow forecasting in various regimes.

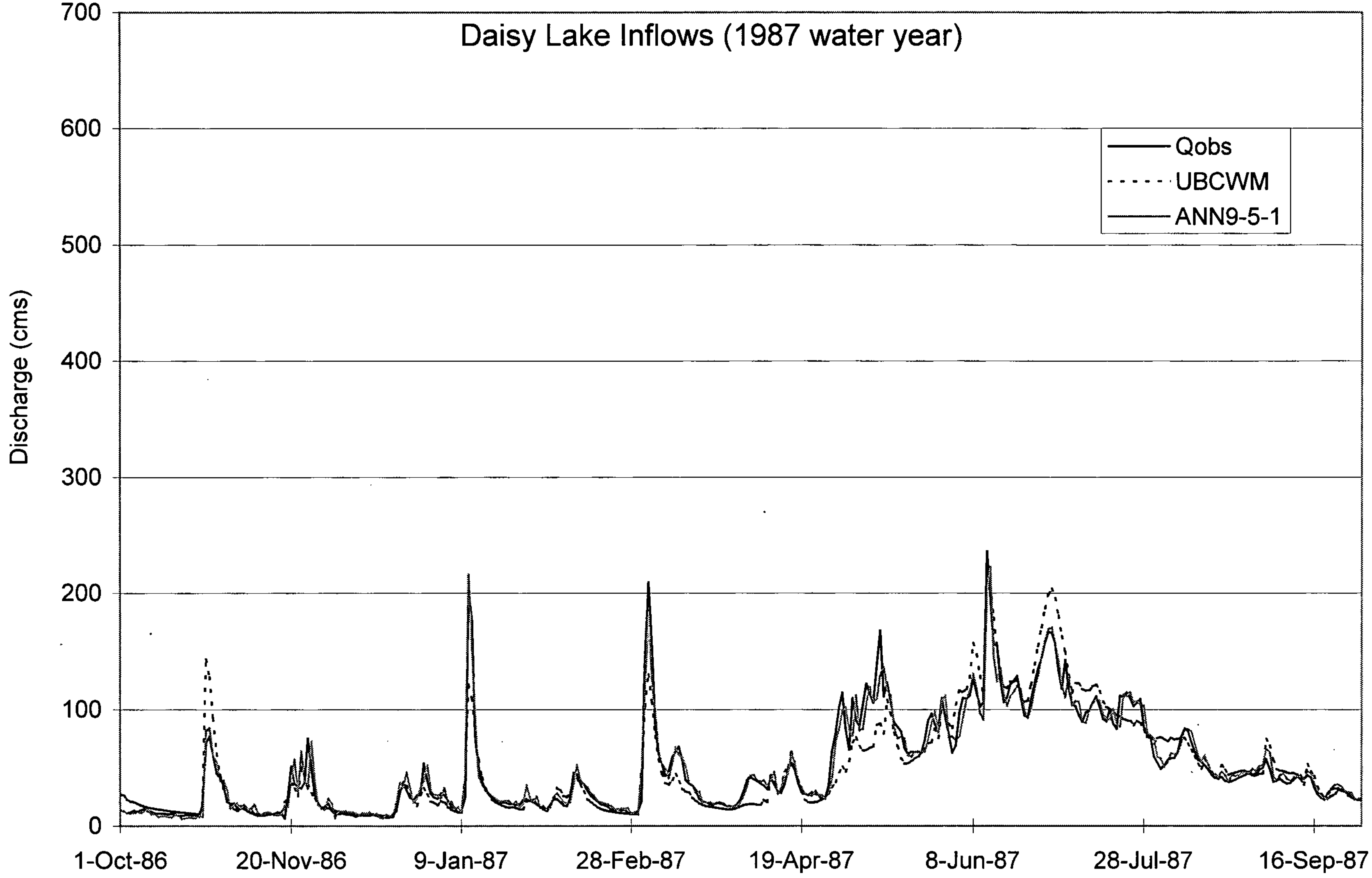

Figure 4.19 Hydrograph comparison (validation, 1987)

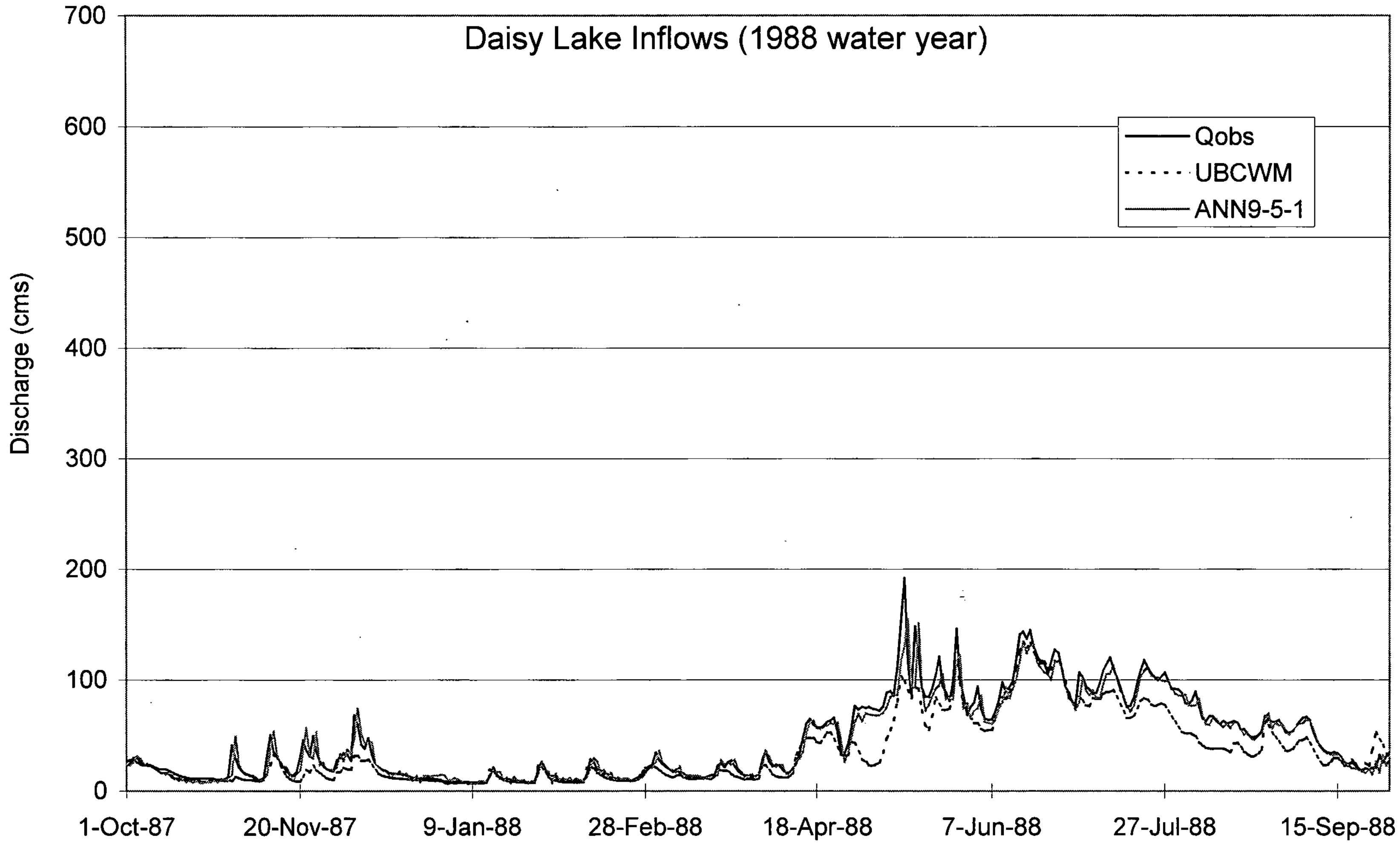

Figure 4.20 Hydrograph comparison (validation, 1988)

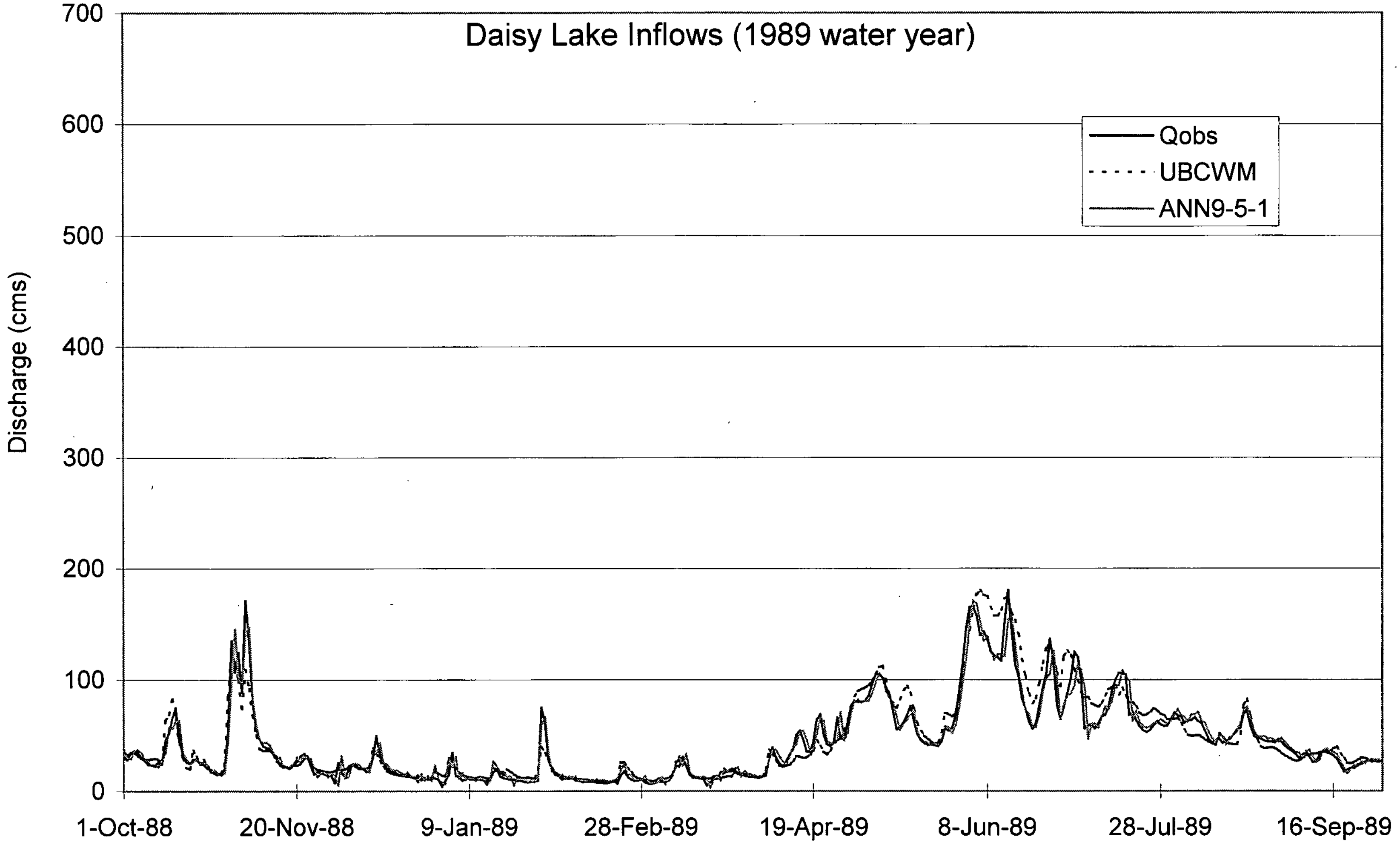

Figure 4.21 Hydrograph comparison (validation, 1989)

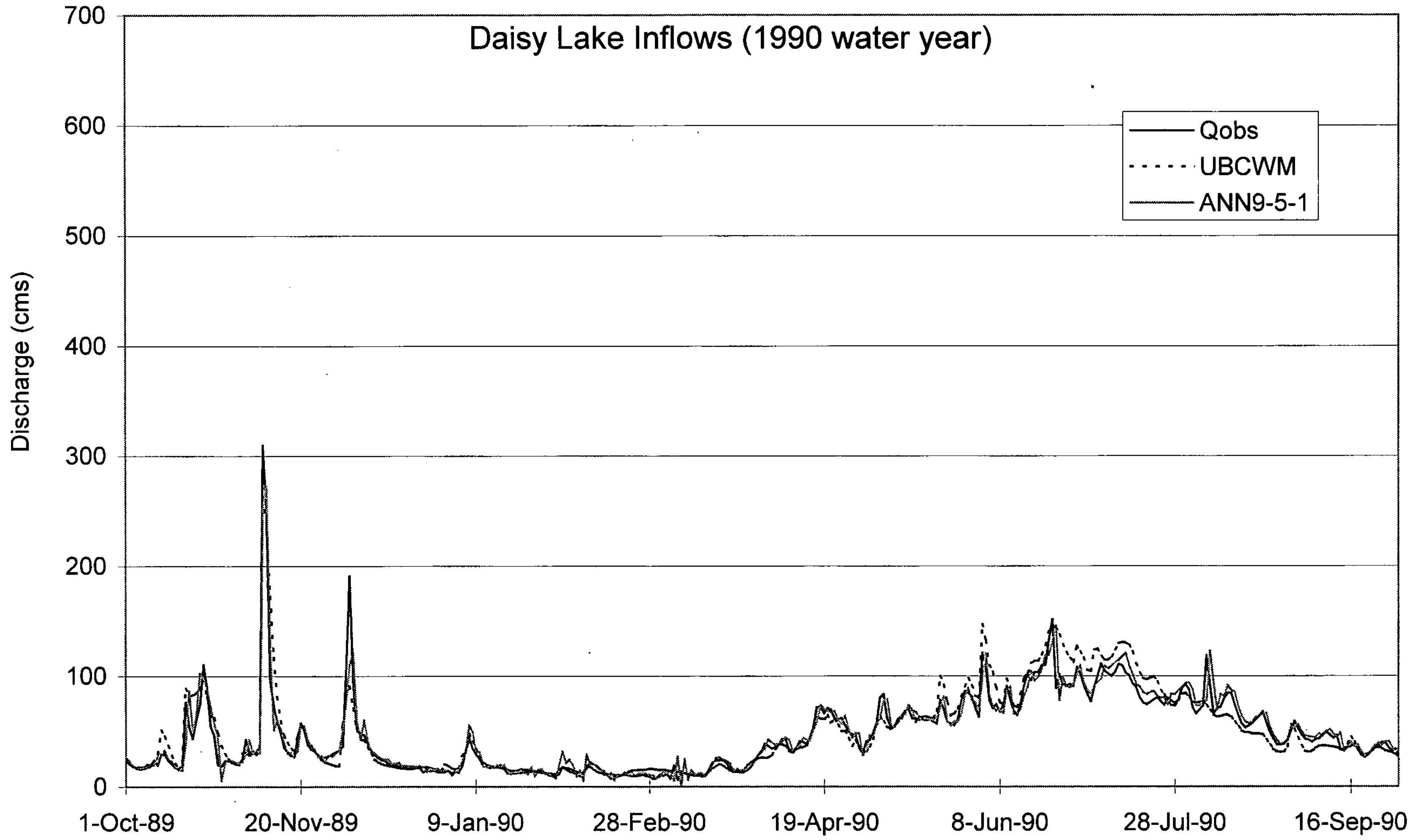

Figure 4.22 Hydrograph comparison (validation, 1990)

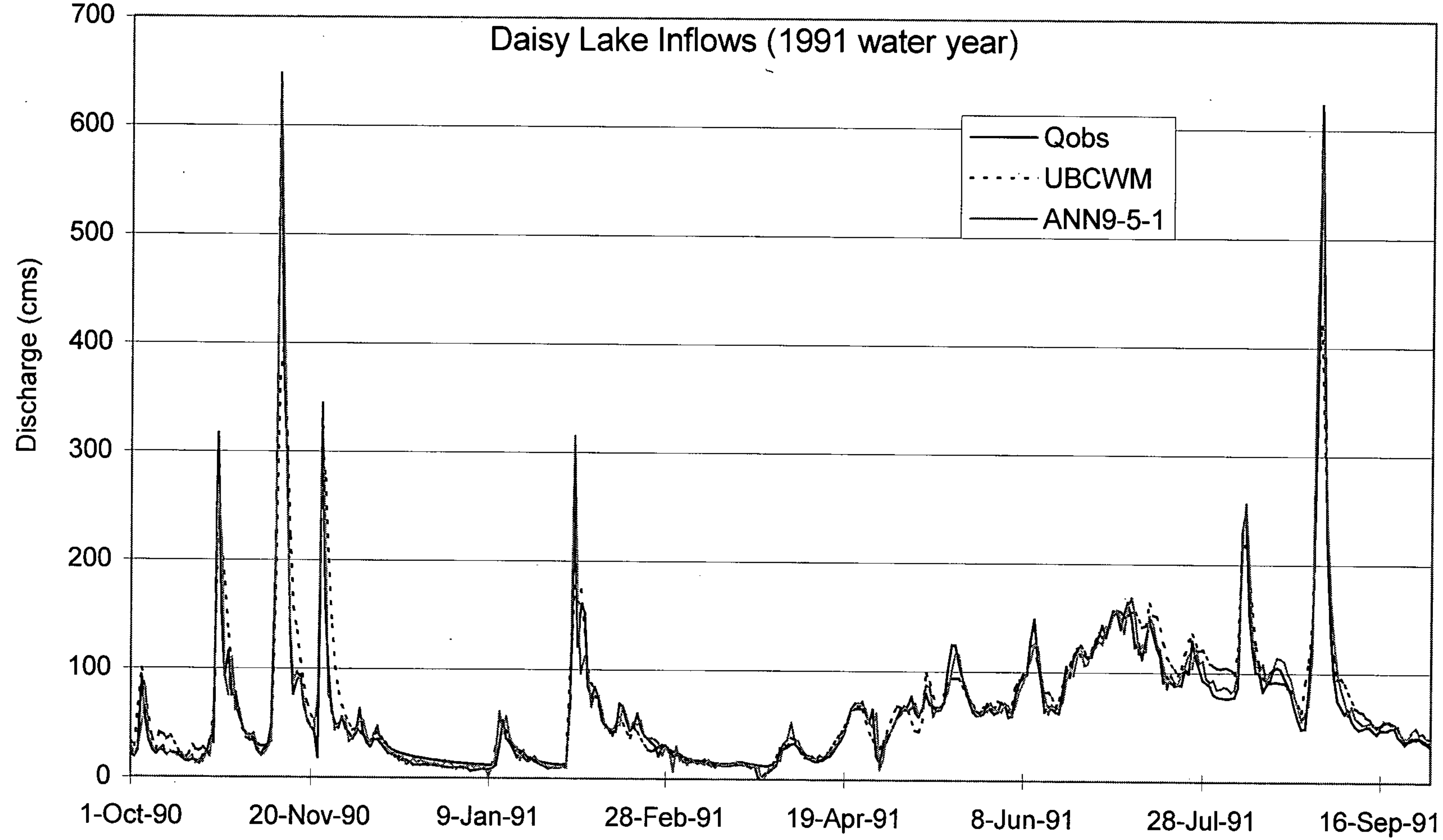

Figure 4.23 Hydrograph comparison (calibration, 1991)

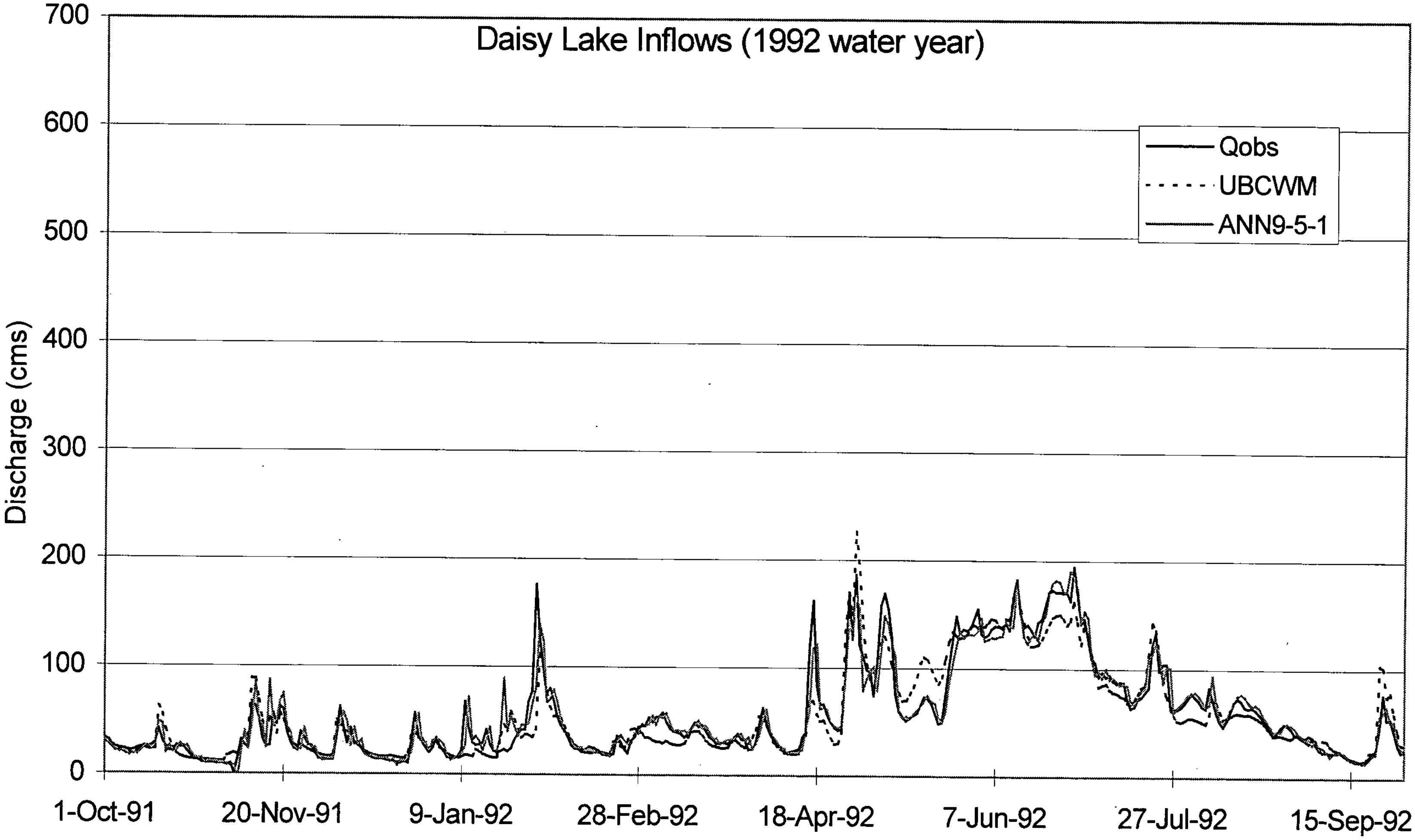

Figure 4.24 Hydrograph comparison (calibration, 1992)

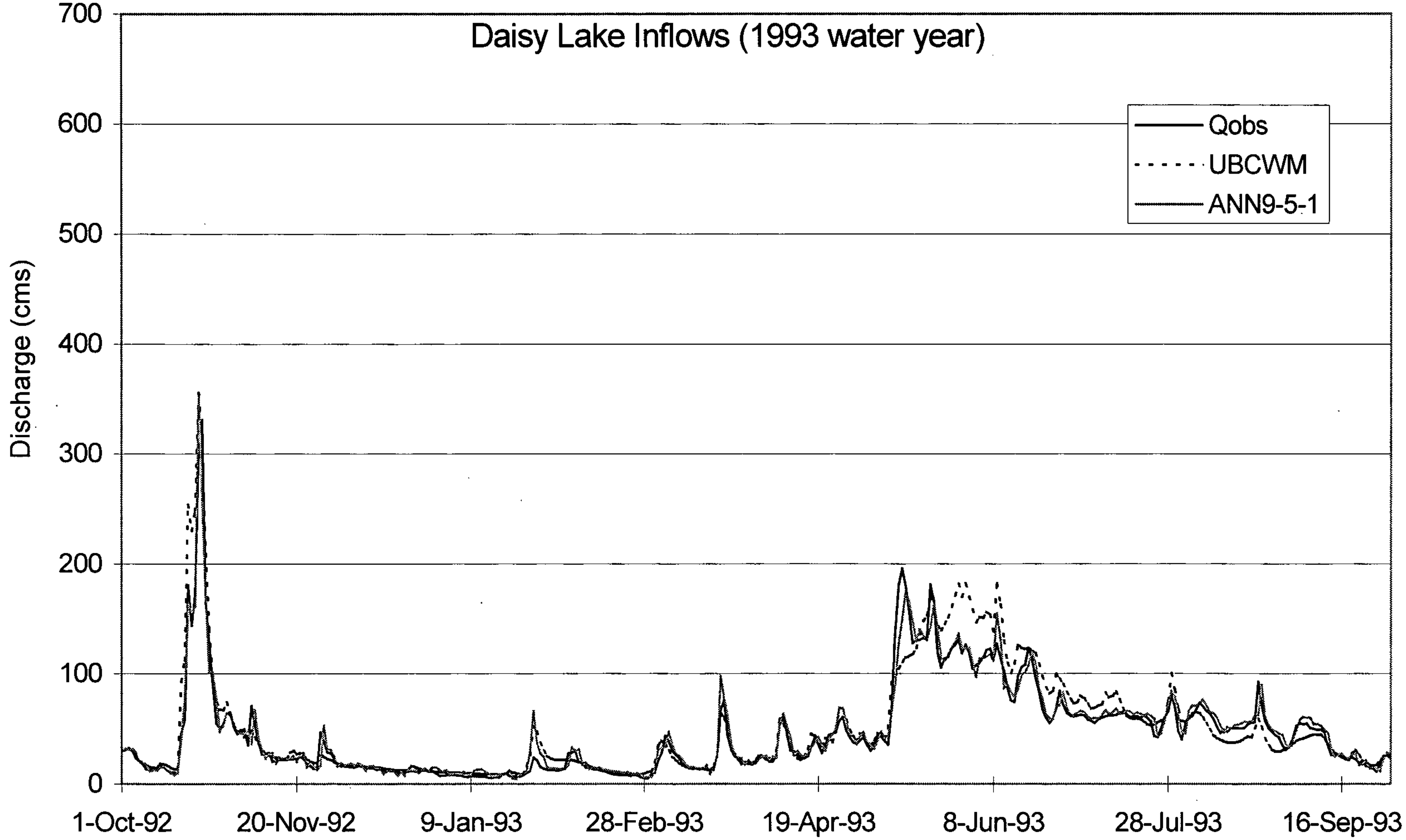

Figure 4.25 Hydrograph comparison (calibration, 1993)

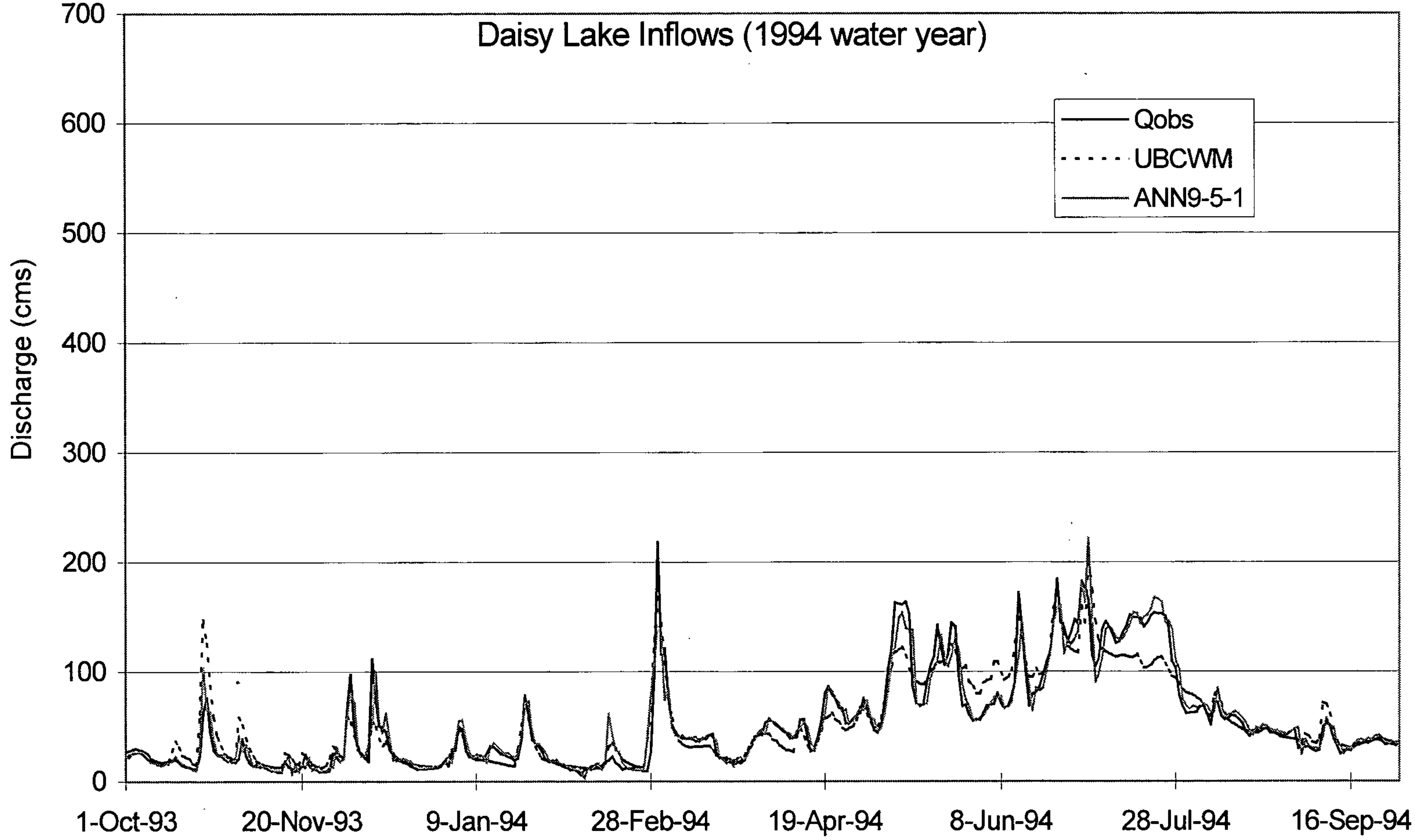

Figure 4.26 Hydrograph comparison (calibration, 1994)

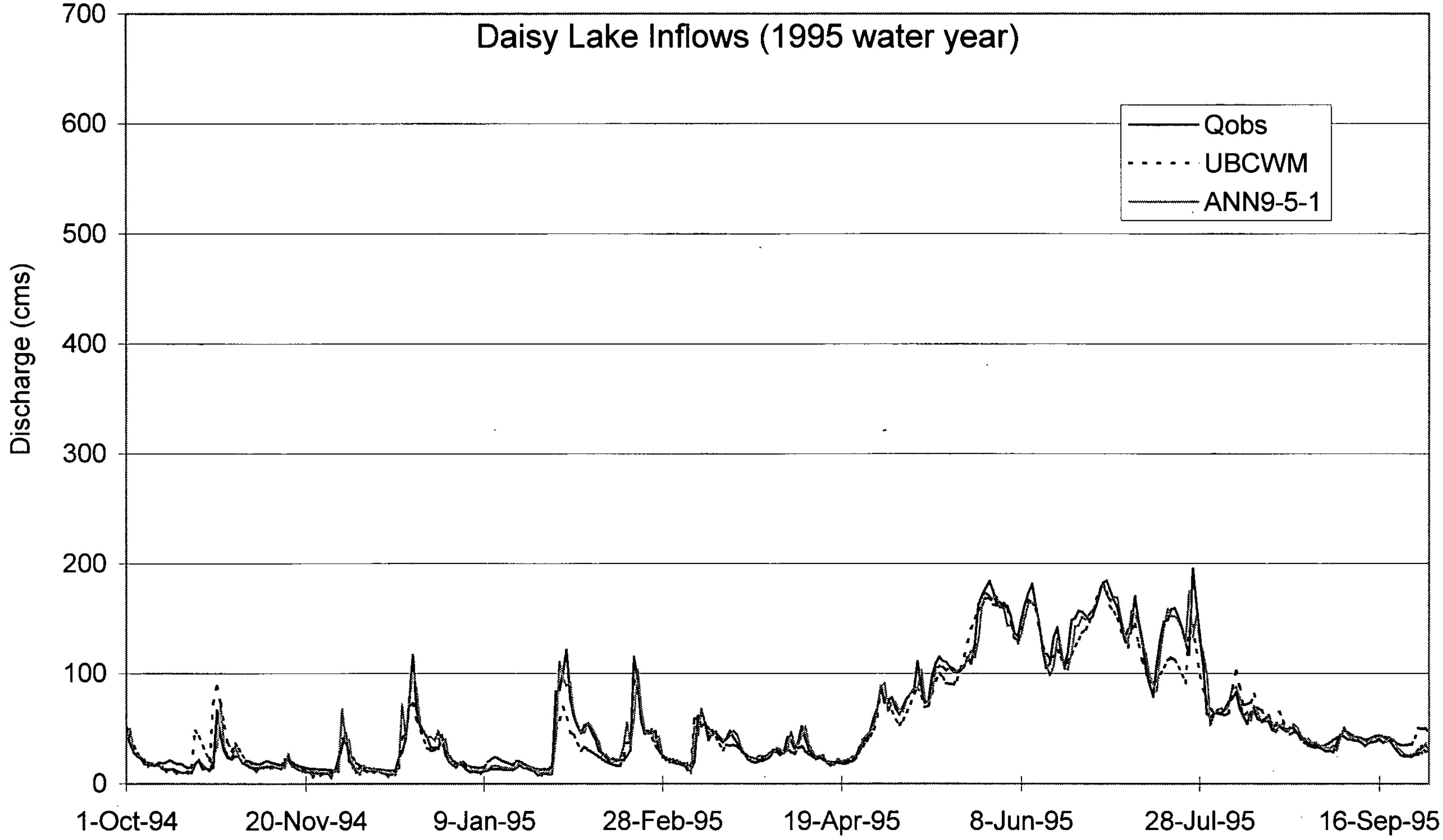

Figure 4.27 Hydrograph comparison (calibration, 1995)

4.2.7 ANN Model Online Test

The ANN9-5-1 model has been implemented in BCH by Frank Weber who is a hydrologist at BCH. All the testing data were retrieved from BCH database. The UBCWM simulation results were obtained from the its River Floods System (RFS), which is operated by BCH hydrologists. Table 4.4 shows a comparison of statistical indices between the ANN9-5-1 model and the UBCWM for data in the period from October 2002 to September 2004. Figures 4.28 and 4.29 are testing results for 2003 and 2004 Daisy Lake inflows.

Table 4.4 ANN9-5-1 model online testing results

YEAR	Duration	Mean of Obs.	RMSE (cms)		MAE (cms)		CE	
	(YY/MM)	(cms)	ANN	UBCWM	ANN	UBCWM	ANN	UBCWM
2003	2002/10-03/09	45.63	9.31	16.76	5.40	12.28	0.93	0.78
2004	2003/10-04/09	48.86	11.63	22.50	6.75	17.34	0.95	0.82

As illustrated in Table 4.4, the performance of the ANN9-5-1 model in the online test is better than the UBCWM. The *CEs* of the ANN9-5-1 model are all greater than 0.93 in two of the online testing years, while *CEs* of the UBCWM are only 0.78 and 0.82, respectively, which is 0.14 less than that of the ANN9-5-1 model. The RMSEs and MAEs show similar performance difference between the two models. *MAEs* of the ANN9-5-1 model, which are less than half the values of the UBCWM, are about 13% of corresponding annual average discharge. The relatively low ratios between MAE and annual average discharge indicate the high modelling accuracy of the ANN9-5-1 model. The higher performance of ANN9-5-1 model in calibration and validation will guarantee that it will perform better than the UBCWM.

The results listed in Tables 4.3 and 4.4 indicate that the ANN9-5-1 model's performance is consistent in terms of *CE*, *MAE* or *RMSE* in all the three periods: calibration, validation and online testing periods; while the UBCWM performance has certain degradation. The *CEs* of the ANN9-5-1 model in the three periods were 0.95, 0.95 and 0.94, respectively. The *CEs* of the UBCWM decreased from 0.87 (calibration) and 0.84 (validation) to 0.80 in the online testing period. The *MAEs* of the ANN9-5-1 model are consistently 11-14% of the average inflow in the corresponding period. But for the UBCWM, its MAEs were 20% of average flows in calibration and testing periods, and 27% and 35% for the two online testing years. The reason for UBCWM performance deterioration is that the calibration and validation datasets are well quality controlled data and have been verified by various departments in BCH, while the 2003 and 2004 data are real-time data with much less quality control. The UBCWM performance subject to noisy input, consequently, has lower performance in real-time forecast situation. The ANN9-5-1 model is not so sensitive to noisy input (refer to subsection 4.2.6) as UBCWM. It is expected that the ANN9-5-1 model has consistent performance in calibration, validation and real-time forecast.

Figures 2.28 and 2.29, which correspond to Table 4.4, are the ANN9-5-1 and UBCWM hydrograph comparison for 2003 and 2004 water years. The figures graphically show the performance of the two models in various streamflow regimes. The figures show that UBCWM constantly overestimates streamflow in snowmelt season, while the ANN9-5-1 model simulated hydrographs match the observed hydrographs very well. The performance of the two models in the snowmelt season in the two years is the same as that in calibration and validation datasets. The performance of the two models in simulating low flows is the same. The performances of the two models in simulating flashy floods is also similar to the performance during calibration and validation. The UBCWM underestimates the October 2004 flood (Figure 4.29). It also can not simulate well the flood recession limb. Figure 4.29 shows that ANN9-5-1 model's performance on the flashy flood is better than the UBCWM. The model online test results prove the conclusion in subsection 4.2.6.

It can be concluded that (1) the ANN9-5-1 model performance is higher than the UBCWM in model training/calibration, validation and online testing periods; (2) the ANN9-5-1 model adapts to the non-linear rainfall-runoff processes in the case study watershed more than UBCWM; (3) ANN9-5-1 is a reliable model that has been proven by the consistency of model performance in the training, validation and online testing periods; (4) the ANN9-5-1 model is insensitive to noisy input data, therefore, it is more suitable for real-time streamflow forecasting; (5) the watershed water storage trend (*S*) is a good indicator for an ANN model to simulate snowmelt induced streamflow. The successful application of the ANN9-5-1 model in the case study watershed indicates that ANN technique, coupled with the model input selection and proper model training method, is a practical and a good option for BCH's small hydro plants streamflow forecasting.

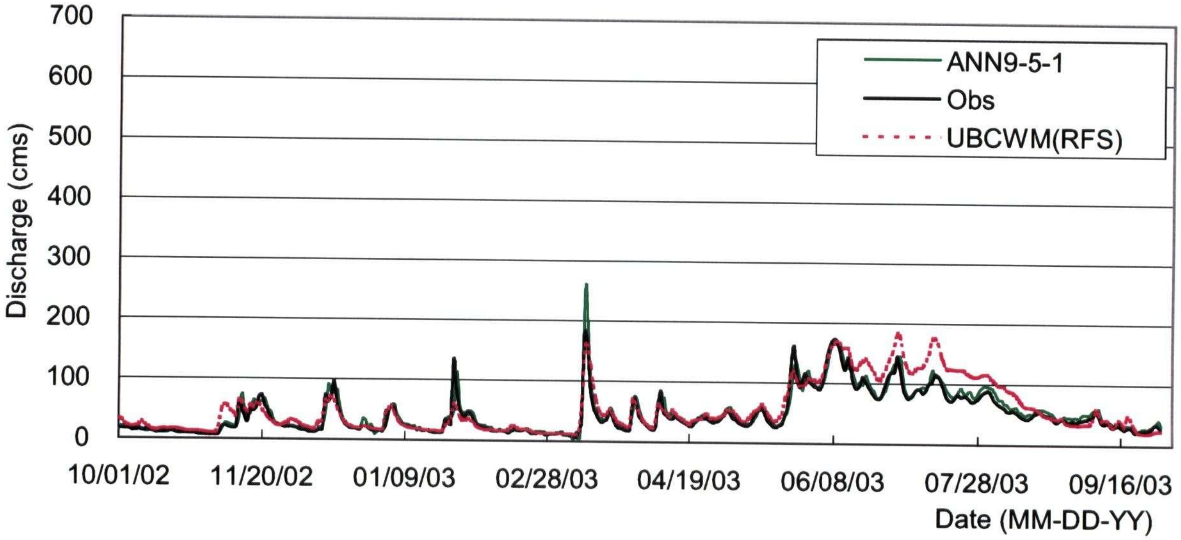

Figure 4.28 ANN9-5-1 and UBCWM online test for 2002-03 Daisy Lake inflow

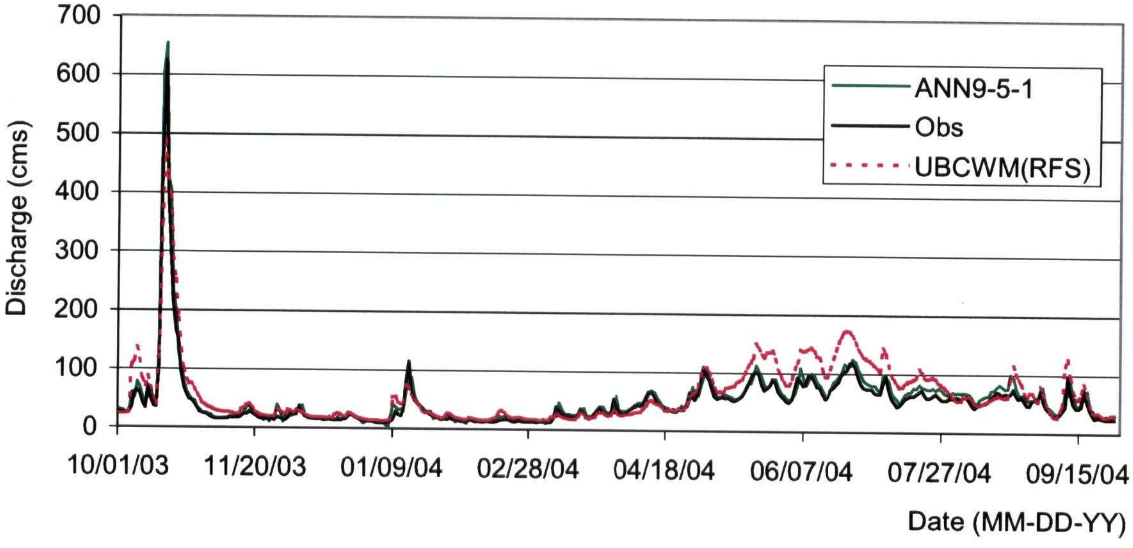

Figure 4.29 ANN9-5-1 and UBCWM online test for 2003-04 Daisy Lake inflow

4.3 Hourly Time Step ANN Model

One of the objectives of the research was to design ANN model(s) to forecast the hourly flood discharge for the next six hours. The most common approach for carrying out such a forecast is to build a one-step ahead forecasting model, and use the model repeatedly for multi-step ahead forecasting. In this research, separate ANN models were built for 1, 2, 4 and 6-hour lead-time flood forecasting. In real-time flood forecasting, a 3-hour lead-time streamflow forecast can be derived by using the 2-hour and 1-hour lead-time models. Because the UBCWM in BCH RFS is a daily time step model and can not provide hourly time step streamflow forecast, a black-box model, the Multi Input Single Output Linear Model (MISOLM), was selected and compared with the 6-hour lead-time ANN model.

4.3.1 Training and Testing Datasets for the ANN Models

The ANN models, which were used for the hourly streamflow forecasting, were trained only by flood events for the purpose of best model performance for flood forecasting rather than low flow forecasting. The inclusion of high and low flow in the training dataset will result in an ANN model that performs poorly for flood forecasting but it could be good for average situation. Thirty flood events in the period October 1986 to September 1998 were chosen to develop the ANN models. All the flood events have a peak flow that exceeds 100m^3/s. Figure 4.30 shows the 30 selected flood events, the 30 floods are plotted on the same figure one after another in a sequence showed in Table 4.5. Table 4.5 lists the start time, end time and peak of each flood event.

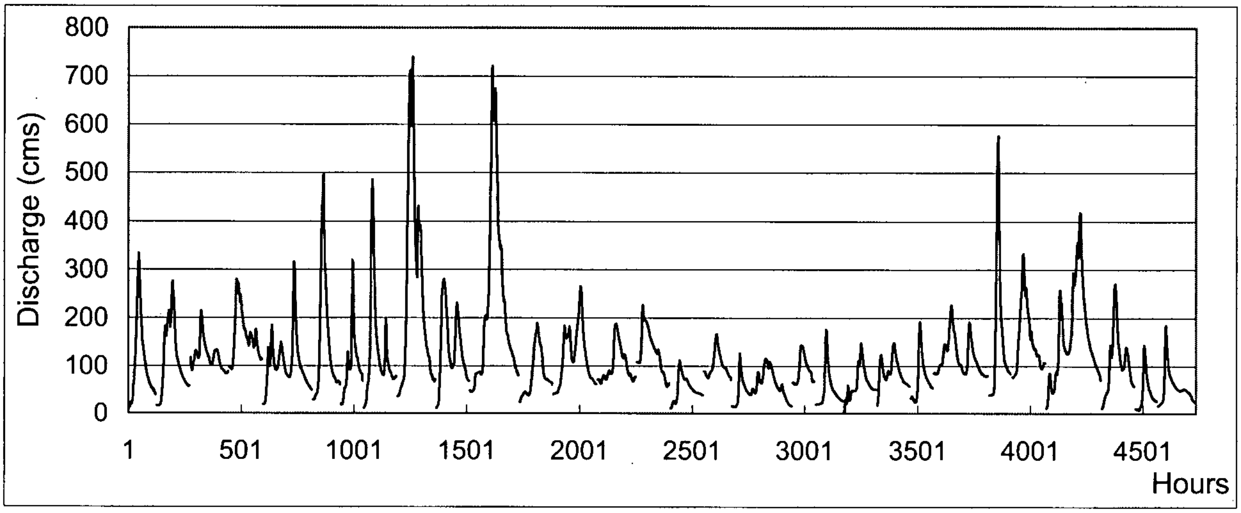

Figure 4.30 Flood events selected for ANN models training and testing

Table 4.5 Flood events selected for ANN models training and testing

	Flood	Begin	End	Peak	Flood	Begin	End	Peak
	No.	mm-dd-yy	mm-dd-yy	(cms)	No.	mm-dd-yy	mm-dd-yy	(cms)
Model training	F1	01-10-87	01-14-87	335	F20	04-25-92	05-02-92	265
	F2	03-03-87	03-08-87	276	F27	06-10-94	06-16-94	188
	F3	05-10-87	05-16-87	387	F33	07-25-95	07-30-95	226
	F4	06-11-87	06-16-87	284	S1	10-25-86	10-30-86	111
	F7	10-31-88	11-08-88	318	S2	05-26-88	05-30-88	166
	F9	11-08-89	11-12-89	496	S5	10-17-89	10-27-89	130
	F10	12-02-89	12-05-89	319	S6	06-02-90	06-05-90	143
	F12	10-24-90	10-29-90	486	S7	12-02-93	12-06-93	174
	F13	11-08-90	11-15-90	**740**	S9	12-17-94	12-22-94	148
	F15	02-01-91	02-06-91	279	S10	01-29-95	02-03-95	150
	F17	08-26-91	09-03-91	722	S11	02-19-95	02-22-95	192
	F19	04-14-92	04-19-92	189				
Model testing	F6	05-10-88	05-19-88	226	F25	02-28-94	03-05-94	270
	F14	11-22-90	11-25-90	578	S3	01-29-89	02-01-89	143
	F16	08-07-91	08-12-91	333	S8	12-09-93	12-15-93	185
	F23	10-18-92	10-27-92	419				

The partition of dataset (streamflow and corresponding input pattern) into training, monitoring and testing datasets is done by a special way. First, flood events were divided into two groups. The first group, which contains the first 23 flood events on Figure 4.30, was used to train the ANN models. The second group, which contains the last seven flood events on Figure 4.30, was used to test the ANN models. The testing floods were chosen by the peak magnitude, occurring time in terms of month and whether they were single or multiple peak events. Second, to utilize the early stop technique in ANN training, the 23 training floods were further separated into training dataset and monitoring dataset by a special method. The method first sorts the combos of streamflow and corresponding input pattern by streamflow. Then the method selects one combo from every five successive combos as a monitoring dataset, and the remaining combos were used as training dataset.

Although streamflows are continuous time series, the dataset partition method was still applicable due to the inherent nature of the feedforward ANN. Before discussing the inherent nature of the ANN models, several features of the conventional hydrological models are presented as below for comparison with that of ANN models. The future streamflow (e.g. next time step) in a conventional hydrological model (e.g. UBCWM) is calculated by:

$$Q = f(x_1, \cdots, x_n, s_1, \cdots, s_m) \tag{4.9}$$

where $x_1, ..., x_n$ are observed hydrometric data, $s_1, ..., s_m$ are model internal states, each state is calculated by $x_1, ..., x_n$, past state and model parameters. In a model calibration process, the model parameters will be changed/adjusted repeatedly, randomly or iteratively depending on the calibration method. As a result, the input pattern to a conventional hydrological model (Equation 4.9) at a time step can only be fully determined during the calibration process. Therefore, the hydrological model input time series (e.g. precipitation) must be kept unchanged in its original sequence during all the calibration process. In an ANN model, the future streamflow is determined by:

$$Q = f(x_1, \cdots, x_n) \tag{4.10}$$

where $x_1, ..., x_n$ is an input pattern formed by observed precipitation, evaporation, temperature, past streamflow or a combination of these (e.g. water storage trend *S*). The function (Equation 4.10) has a simple form, and its arguments have nothing to do with model parameters. All the ANN inputs in the pattern have nothing to do with past model output information (e.g. the ANN9-5-1 model, Equation 4.8). Furthermore, all the ANN model inputs have nothing to do with model parameters. Once an ANN model (Equation 4.10) has been determined, its input pattern at any time step is known before the model training (calibration) process. The ANN calibration can be thought of as a kind of multivariate regression analysis. Therefore, we could rearrange and sort the observed streamflows and their corresponding input patterns without losing any information related to the runoff calculation. Figure 4.31 shows the streamflow of training, monitoring and testing datasets. The dataset partition method makes patterns, in terms of streamflow magnitude, evenly distributed in training and monitoring datasets. The testing dataset keeps its original time series form for the validation hydrograph comparison.

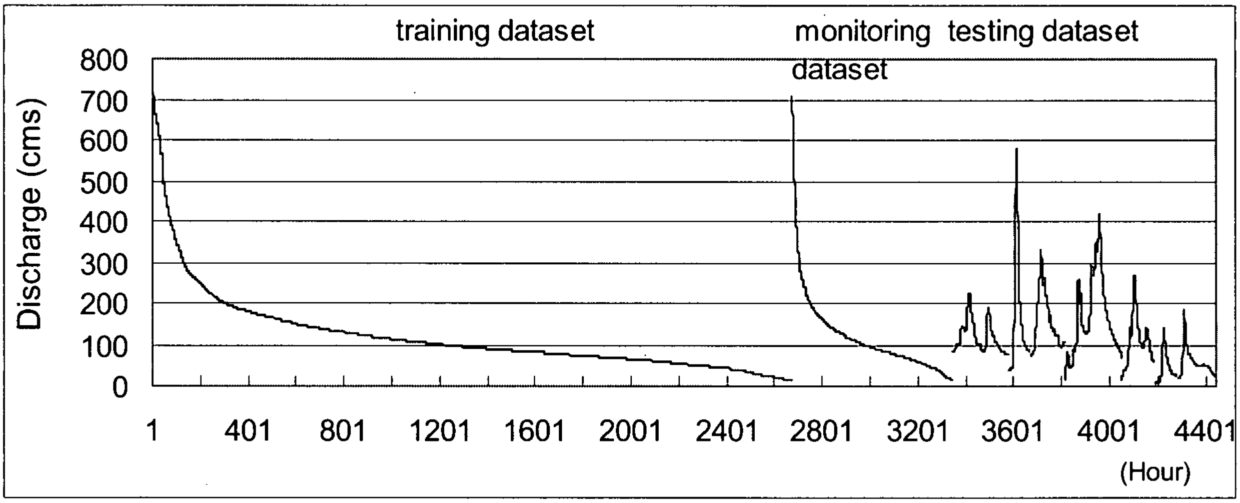

Figure 4.31 The partition of streamflow into training, monitoring and testing dataset

4.3.2 Hourly Time Step ANN Models

Based on the ANN theory and its application for daily streamflow forecasting, four hourly time step ANN models were developed. The models used past and future hydrometric and climate information and watershed water storage trend as inputs. According to the current weather prediction techniques, the estimated average precipitation or temperature in several time steps (future several hours) is obviously more accurate than the value at individual finer time steps.

Therefore all the hourly time step ANN models used an average value for the future one to six hours. Because the hourly temperature change is usually slower than that of precipitation, and for the sake of simplicity, all ANN models used average temperature for both the future and the past.

4.3.2.1 One-hour lead-time ANN model (ANN-1)

The one hour lead-time ANN model is shown by Figure 4.32 and Equation 4.11. The model takes advantage of the past streamflow input indicators Q_{t-1} and Q_{t-2}, because the coefficient of correlation between Q_{t+1} and Q_{t-1} is 0.96. The rainfall-flood response (the main precipitation to flood peak) time in this watershed is about six hours, the precipitation at *t-4* and *t-5* are considered as good input indicators. Subsequent precipitations are not considered as an input for the model, because (1) the past streamflow indicators already contain or summarize most of past information; and (2) the model should be made as simple as possible. The precipitation in the past four hours will affect the magnitude of future streamflow, and for the sake of simplicity, the model uses the average precipitation ($\overline{P}_{t,t-3}$) in the past four hours as input indicator. The water storage trend (S_{t-1}) will definitely feed the ANN models with the soil moisture information.

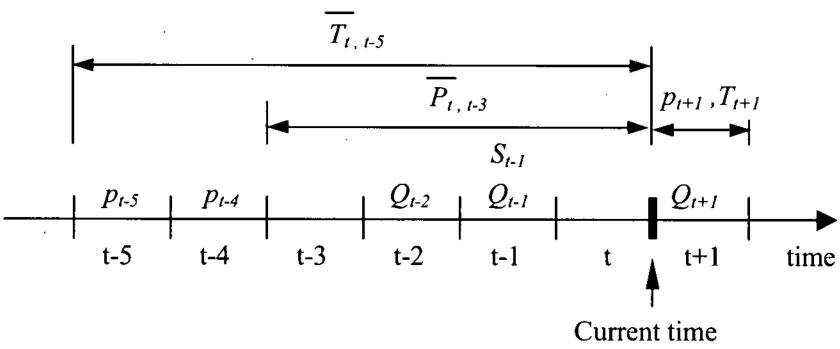

Figure 4.32 One-hour lead time ANN model (ANN-1) structure

Therefore, the forecasted streamflow Q_{t+1} for time step *t+1* can be expressed as:

$$Q_{t+1} = f\left(p_{t+1}, \overline{P}_{t,t-3}, p_{t-4}, p_{t-5}, T_{t+1}, \overline{T}_{t,t-5}, Q_{t-1}, Q_{t-2}, S_{t-1}\right) \tag{4.11}$$

Where:

t Time step;

Q_t Quality controlled Daisy Lake hourly inflow (m^3/s);

p_t Hourly precipitation (mm) at Upper Cheakamus station (CMU);

$\overline{P}_{t,t-3} = \frac{1}{4}\sum_{i=0}^{3} p_{t-i}$ Average precipitation at CMU in the past 4 hours (mm);

T_{t+1} Average temperature at CMU in future 1 hours (℃). Its value is estimated in real-time forecasting;

$\overline{T}_{t,t-5} = \frac{1}{6}\sum_{i=0}^{5} T_{t-i}$ Average temperature at CMU in past 6 hours (℃);

S_{t-1} Hourly watershed water storage trend (mm). Refer to Equation 4.6.

The cross-correlation between Q_{t+1} and precipitation and past streamflows are listed below:

	p_{t-5}	p_{t-4}	$\overline{P}_{t,t-3}$	p_{t+1}	Q_{t-2}	Q_{t-1}
Q_{t+1}	0.48	0.47	0.44	0.3	0.93	0.96

The cross-correlation between Q_{t+1} and past streamflow indicators, Q_{t-1} and Q_{t-2}, is as high as 0.93 and 0.96. The high cross-correlation values indicate that past streamflows, Q_{t-1} and Q_{t-2}, are strong indicators for the ANN model, and consequently secure the high modelling accuracy. The cross-correlation between Q_{t+1} and precipitation indicators vary from 0.3 to 0.48. When considering the short time step and the indirect relationship between streamflow and precipitation, the values of cross-correlation between Q_{t+1} and precipitation are still acceptable.

4.3.2.2 Two-hour lead time ANN model (ANN-2)

The two-hour lead-time ANN model is shown in Figure 4.33 and Equation 4.12.

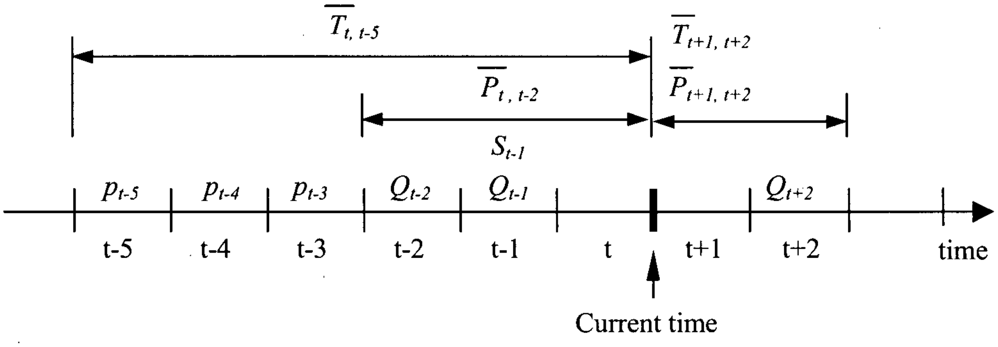

Figure 4.33 Two-hour lead time ANN model (ANN-2) structure

The forecasted streamflow Q_{t+2} for time step $t+2$ can be expressed as:

$$Q_{t+2} = f\left(\overline{P}_{t+1,t+2}, \overline{P}_{t,t-2}, p_{t-3}, p_{t-4}, p_{t-5}, \overline{T}_{t+1,t+2}, \overline{T}_{t,t-5}, Q_{t-1}, Q_{t-2}, S_{t-1}\right) \tag{4.12}$$

Where:

$\overline{P}_{t+1,t+2} = \frac{1}{2}\sum_{i=1}^{2} p_{t+i}$ Average precipitation at CMU in the future 2 hours (mm);

$\overline{P}_{t,t-2} = \frac{1}{3}\sum_{i=0}^{2} p_{t-i}$ Average precipitation at CMU in the past 3 hours (mm);

$\overline{T}_{t+1,t+2} = \frac{1}{2}\sum_{i=1}^{2} T_{t+i}$ Average temperature at CMU in the future 2 hours (℃);

$\overline{T}_{t,t-5} = \frac{1}{6}\sum_{i=0}^{5} T_{t-i}$ Average temperature at CMU in the past 6 hours (℃).

The cross-correlation between Q_{t+2} and precipitation and past streamflows are:

	p_{t-5}	p_{t-4}	p_{t-3}	$\overline{P}_{p3}$	$\overline{P}_{f2}$	Q_{t-2}	Q_{t-1}
Q_{t+2}	0.49	0.48	0.47	0.45	0.33	0.89	0.93

4.3.2.3 Four-hour lead time ANN model (ANN-4)

The four-hour lead time ANN model is shown in Figure 4.34 and Equation 4.13.

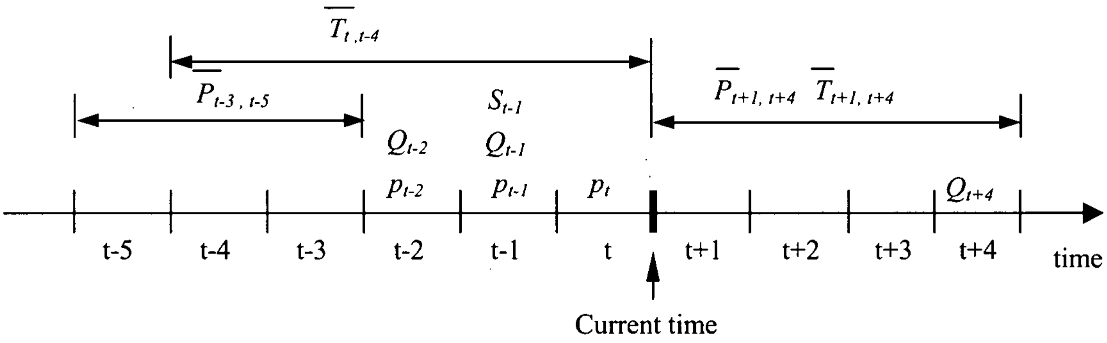

Figure 4.34 Four-hour lead time ANN model (ANN-4) structure

The forecasted streamflow Q_{t+4} for time step *t+4* can be expressed as:

$$Q_{t+4} = f\left(\overline{P}_{t+1,t+4}, p_t, p_{t-1}, p_{t-2}, \overline{P}_{t-3,t-5}, \overline{T}_{t+1,t+4}, \overline{T}_{t,t-4}, Q_{t-1}, Q_{t-2}, S_{t-1}\right) \tag{4.13}$$

where:

$\overline{P}_{t+1,t+4} = \frac{1}{4}\sum_{i=1}^{4} p_{t+i}$ Average precipitation at CMU in the future *4* hours (mm);

$\overline{P}_{t-3,t-5} = \frac{1}{3}\sum_{i=3}^{5} p_{t-i}$ Average precipitation at CMU in hours *t-3* to *t-5* (mm);

$\overline{T}_{t+1,t+4} = \frac{1}{4}\sum_{i=1}^{4} T_{t+i}$ Average temperature at CMU in the future *4* hours (℃);

$\overline{T}_{t,t-4} = \frac{1}{5}\sum_{i=0}^{4} T_{t-i}$ Average temperature at CMU in the past *5* hours (℃).

The cross-correlation between Q_{t+4} and precipitation and past streamflows are:

	$\overline{P}_{p3}$	p_{t-2}	p_{t-1}	p_t	$\overline{P}_{f4}$	Q_{t-2}	Q_{t-1}
Q_{t+4}	0.54	0.48	0.47	0.45	0.4	0.81	0.85

4.3.2.4 Six-hour lead time ANN model (ANN-6)

The six-hour lead-time ANN model is shown in Figure 4.35 and Equation 4.14. The model is characterized by its long lead-time that almost matches the length of the rainfall-flood response time (the time from main precipitation to flood peak). This means that the six-hour lead-time (*t+6*) streamflow is determined by the current watershed state and the future precipitation. The water storage trend (S_{t-1}) and the streamflow (Q_{t-1}) are obvious good indicators of the current watershed state, while the additional streamflow information (Q_{t-1} and Q_{t-2}) were used to inform the ANN model about streamflow trends. The precipitation data were chosen as input indicators due to the following considerations:

- The streamflow indicators and the water storage trend indicator do not totally summarize the concurrent precipitation information;
- Past streamflows do not show the ANN model long term streamflow trend due to the 7-hours temporal difference between Q_{t+6} and Q_{t-1};
- Due to the rainfall runoff lag time, precipitation is a good indicator that shows the ANN model short term trend.

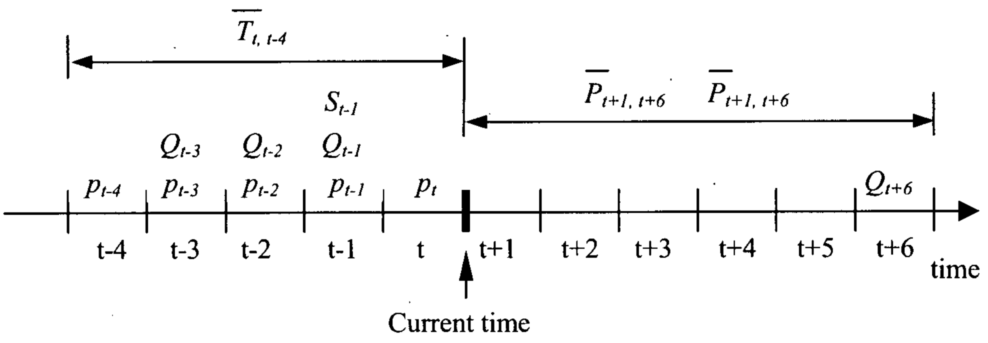

Figure 4.35 Six-hour lead time ANN model (ANN-6) structure

The forecasted streamflow Q_{t+6} for time step $t+6$ can be expressed as:

$$Q_{t+6} = f\left(\overline{P}_{t+1,t+6}, p_t, p_{t-1}, p_{t-2}, p_{t-3}, p_{t-4}, \overline{T}_{t+1,t+6}, \overline{T}_{t,t-4}, Q_{t-1}, Q_{t-2}, Q_{t-3}, S_{t-1}\right) \qquad (4.14)$$

where:

$\overline{P}_{t+1,t+6} = \frac{1}{6}\sum_{i=1}^{6} p_{t+i}$ Average precipitation at CMU in the future 6 hours (mm);

$\overline{T}_{t+1,t+6} = \frac{1}{6}\sum_{i=1}^{6} T_{t+i}$ Average temperature at CMU in the future 6 hours (℃);

$\overline{T}_{t,t-4} = \frac{1}{5}\sum_{i=0}^{4} T_{t-i}$ Average temperature at CMU in the past 5 hours (℃).

The cross-correlation between Q_{t+6} and precipitation and past streamflow are:

	p_{t-4}	p_{t-3}	p_{t-2}	p_{t-1}	p_t	$\overline{P}_{f6}$	Q_{t-3}	Q_{t-2}	Q_{t-1}
Q_{t+6}	0.49	0.50	0.50	0.49	0.48	0.46	0.66	0.71	0.76

The cross-correlations between Q_{t+6} and past streamflow (Q_{t-3}, Q_{t-2}, Q_{t-1}) vary from 0.66 to 0.76. They are obviously smaller than the cross-correlation between Q_{t-1} and Q_{t+1} (0.96). This will definitely determine that the modelling accuracy of 6-hour lead-time ANN model is lower that that of 1-hour lead-time ANN model.

4.3.3 Hourly Time Step ANN Models Training Results

The hourly time step ANN models were trained by the same techniques and procedures as what have been used in the daily time step ANN model. Table 4.6 lists the statistics of the training results for the 1, 2, 4 and 6-hour lead-time models. The statistics were calculated according to the training dataset. The optimal number of hidden nodes for the models was three or four. The modelling error, in terms of *RMSE* and *MAE* statistics, increases with the increase in lead-time. *RMSEs* on the training dataset vary between 5.98 and 17.4 m^3/s corresponding to 1 and 6-hour lead-time models, respectively. The *MAEs* on training dataset varied between 3.08 and 10.3 m^3/s corresponding to 1 and 6-hour lead-time models, respectively. The coefficients of model efficiency (*CE*) for the four models are greater than 0.97, and increase with the decrease in lead-time. The relatively high *CE* values illustrate the good performance of the ANN models on the training dataset. Recall that the training dataset was partitioned into two parts by a special method. The training dataset was restored to its original time series form, namely the observed hydrograph. Figures 4.36 to 4.39 show the comparison between simulated and observed hydrographs for 1, 2, 4 and 6-hour lead-time models. The figures indicate that ANN recognizes the relationship between the input indicators and streamflow. It can be observed that the simulated streamflow matches the observed streamflow very closely.

Table 4.6 Hourly time step model training results

ANN model	Lead time	ANN architecture	Mean of Qobs (cms)	RMSE (cms)	CE	R	MAE (cms)	Volume error (%)
ANN-1	1hr	9-3-1	120	5.98	0.996	0.998	3.08	-0.25
ANN-2	2hr	10-4-1	121	9.04	0.992	0.996	4.85	0.43
ANN-4	4hr	10-4-1	123	14.5	0.979	0.990	8.19	0.09
ANN-6	6hr	12-4-1	123	17.4	0.970	0.985	10.31	0.05

Note 1: Format of ANN architecture is I-H-O. I is the number of input nodes, H is the number of hidden nodes, O is the number of output nodes.

Note 2: The difference in "Mean of Qobs" value is caused by the size of training dataset. Although all the models are tested by the same 7 testing floods, the size of testing dataset for each model is slightly different due to the difference in input pattern. Given a flood time series and corresponding precipitation and temperature, when forming ANN input pattern, several streamflow data at both end of flood can not be used to train ANN models.

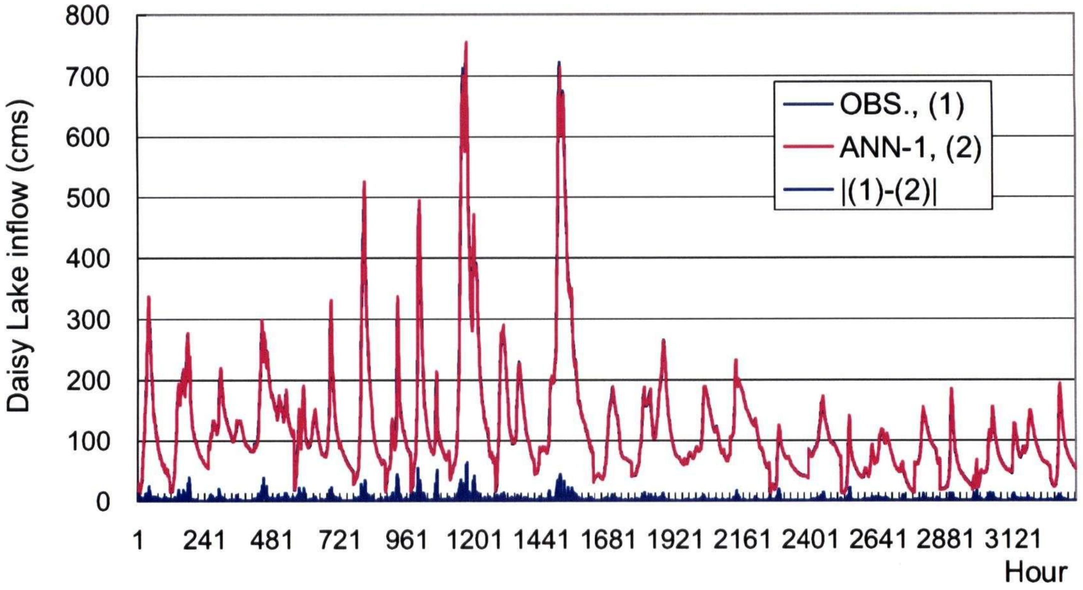

Figure 4.36 ANN-1 performance on 23 training flood events

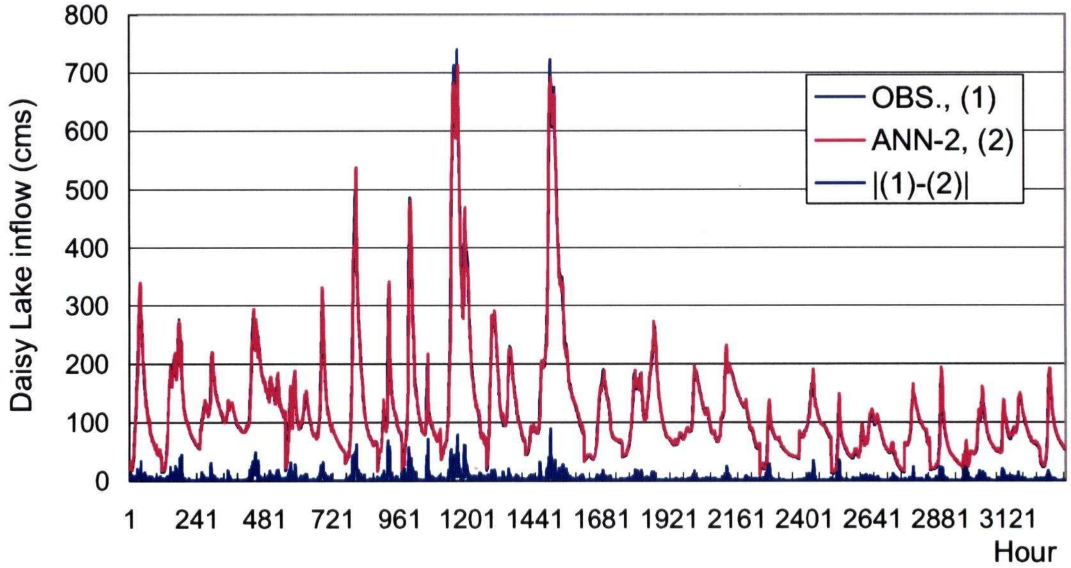

Figure 4.37 ANN-2 performance on 23 training flood events

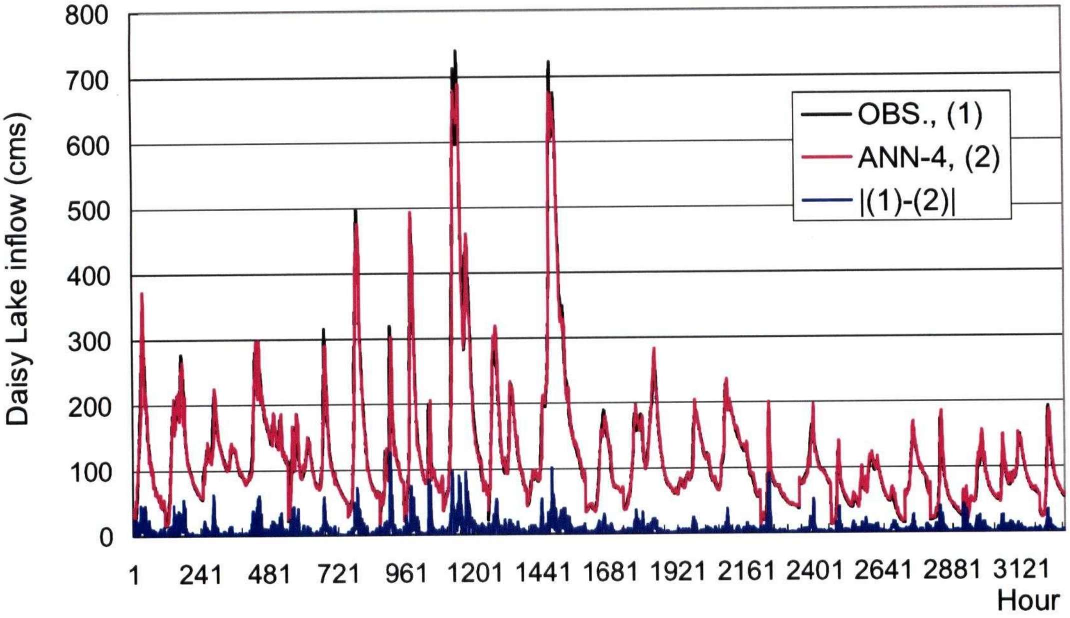

Figure 4.38 ANN-4 performance on 23 training flood events

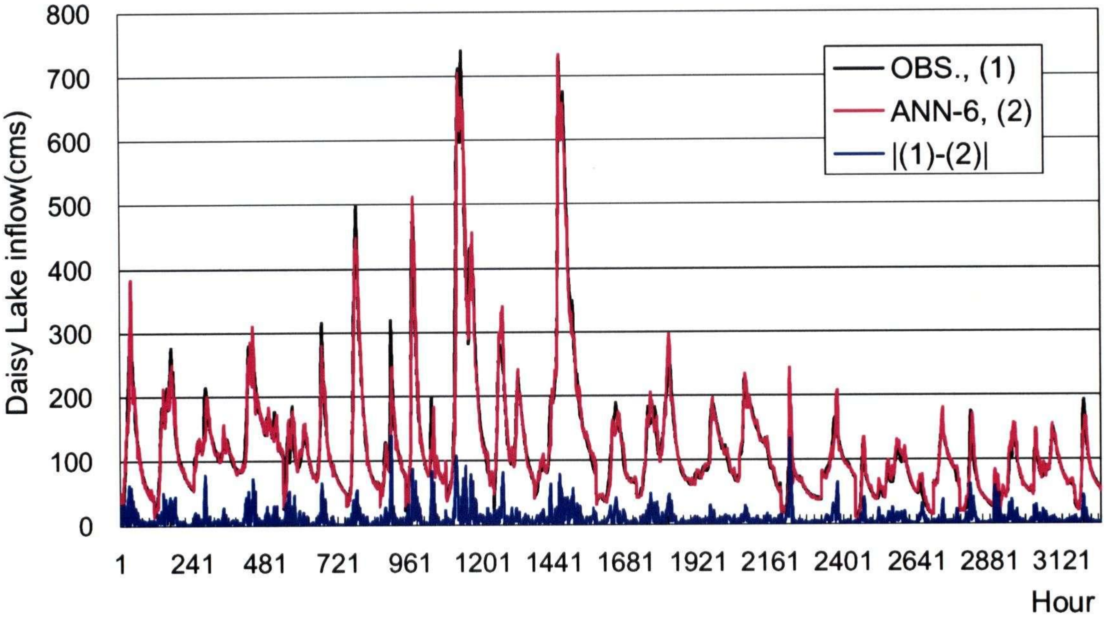

Figure 4.39 ANN-6 performance on 23 training flood events

Table 4.7 Hourly time step ANN models testing results

ANN Model	Lead time	ANN architecture	Mean of obs.(cms)	RMSE (cms)	CE	R	MAE (cms)	Volume error%
ANN-1	1hr	9-3-1	123	5.29	0.996	0.998	3.05	-0.14
ANN-2	2hr	10-4-1	124	8.12	0.991	0.996	4.83	0.53
ANN-4	4hr	10-4-1	123	14.0	0.973	0.987	8.60	1.3
ANN-6	6hr	12-4-1	125	18.8	0.951	0.977	11.9	1.89

Table 4.7 shows the test results for one, two, four and six-hour lead-time ANN models. The optimal number of hidden nodes is three or four for those models. The *RMSE* and *MAE* statistics increase with the increase of lead-time, with the six-hour lead time model having the biggest modelling error of *RMSE* 18.8 m^3/s, or 15.0% of mean observed streamflow. The models' *CE* and *R* statistics decrease with the increase of lead-time, *CEs* for all of the models are greater than 0.95. The high values of *CEs* illustrate the good performance of the ANN models on the test floods. Figures 4.40 to 4.43 show the good agreement between observed and simulated hydrographs on the seven testing flood events. The simulated hydrograph matches the observed hydrograph very well for floods of different magnitudes. It also indicates that the ANN models can give a good match for the flood rise and recession limbs, and for single and multiple peak events.

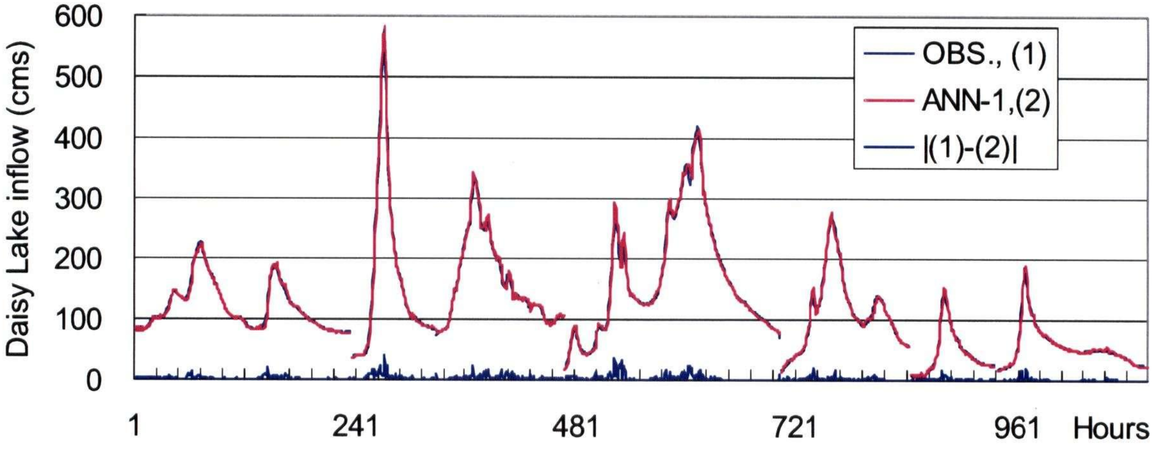

Figure 4.40 ANN-1 performance on seven testing flood events

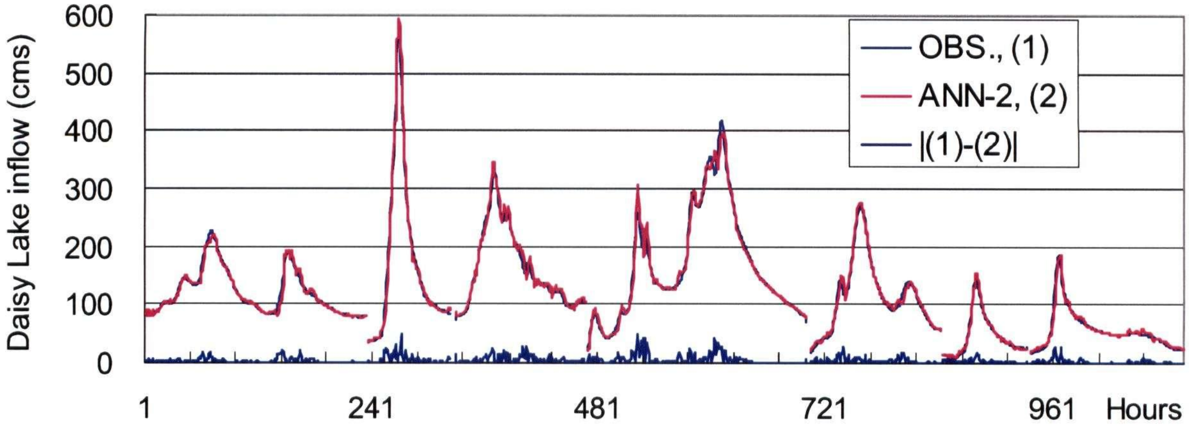

Figure 4.41 ANN-2 performance on seven testing flood events

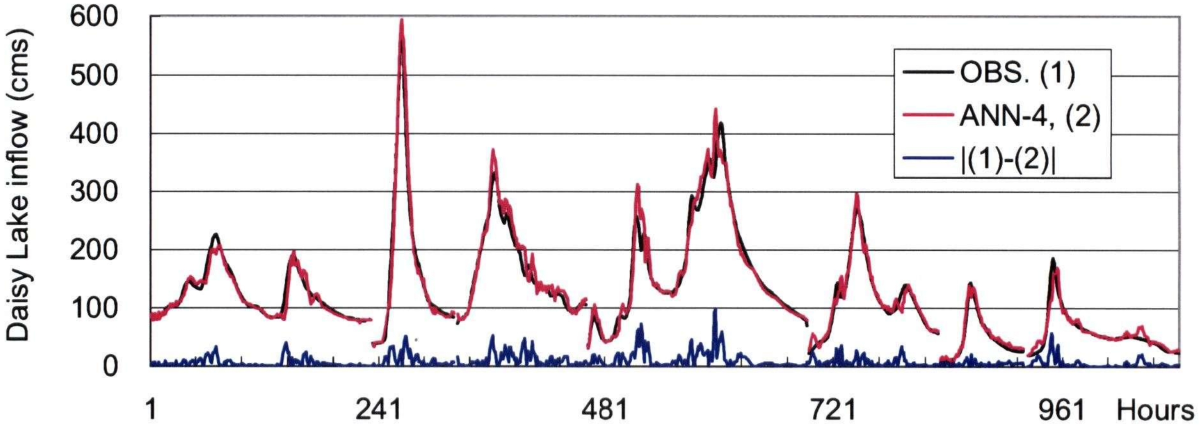

Figure 4.42 ANN-4 performance on seven testing flood events

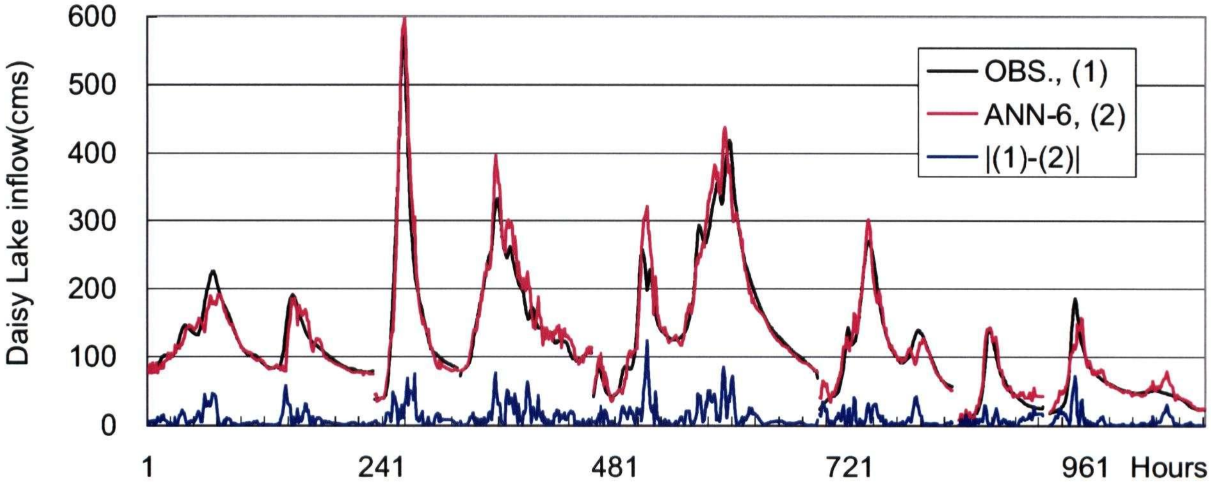

Figure 4.43 ANN-6 performance on seven testing flood events

4.3.4 Performance Comparison Between ANN and MISOLM

In this research, the Multi-Input Single Output Linear Model (MISOLM, Kachroo and Liang, 1992, refer to Appendix A) was selected as a reference model and compared with the six-hour lead-time ANN model (ANN-6).

To compare the MISOLM with the ANN-6 model, the input indicators used by ANN-6 model were also used by MISOLM as initial model inputs. In the MISOLM calibration, a trial-and-error process was adapted to refine the input indicators. The effort was focused on finding the best combination of the input indicators. The final MISOLM input indicators were the same as the model shown in Equation 4.15 and Figure 4.44. The MISOLM used the same flood events, which were also used by the ANN-6 model, for model calibration and testing.

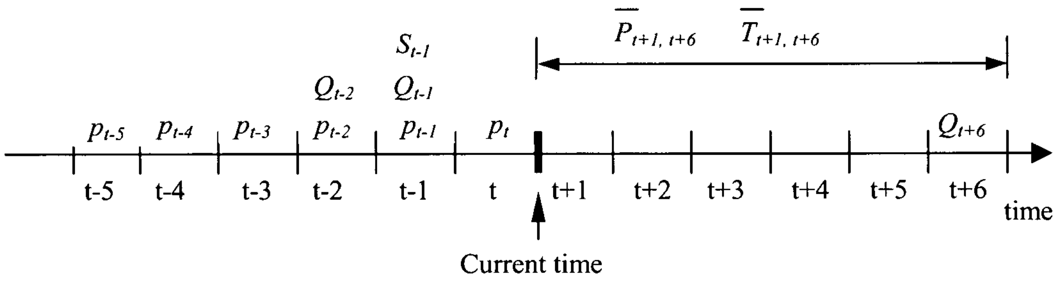

Figure 4.44 MISOLM inputs for 6-hour ahead streamflow forecasting

The forecasted streamflow Q_{t+6} for time step $t+6$ can be expressed as:

$$Q_{t+6} = f\left(p_t, p_{t-1}, p_{t-2}, p_{t-3}, p_{t-4}, p_{t-5}, \overline{P}_{t+1,t+6}, \overline{T}_{t+1,t+6}, Q_{t-1}, Q_{t-2}, S_{t-1}\right) \qquad (4.15)$$

Where t is time step in hour, p is precipitation, Q is streamflow, S is water storage trend and $\overline{P}_{t+1,t+6}$ and $\overline{T}_{t+1,t+6}$ are average precipitation and temperature for the future 6 hours.

Table 4.8 lists the MISOLM model calibration and testing results and the corresponding ANN model results. The MISOLM has average coefficient of model efficiency (*CE*) of 0.93 and 0.90 in calibration and testing periods, respectively. It simulates the flood rising limb, recession limb and timing reasonably well in most test floods (Figure 4.45). The only exception is that the MISOLM underestimates part of the recession limb for the second last floods. The MISOLM testing error (*MAE*) is 17.8m^3/s, which is 14.2% of the average observed streamflow. In general, the MISOLM is an acceptable model for the case study watershed under consideration. The comparison between the ANN-6 and the MISOLM performance shows that the ANN-6 model is better than the MISOLM in all statistics. The *MAE* of the ANN-6 model is less than that of the MISOLM by about 5.9m^3/s, and consequently, *CE* of the ANN-6 model is 0.05 higher than MISOLM. Figure 4.45 compares the hydrographs of test floods generated by the two models. The ANN-6 model generated hydrograph is more in agreement with the observed hydrograph than that generated by the MISOLM, particularly for the last two floods. The good performance of the ANN-6 model on all testing floods indicates that the ANN-6 model adapts to flood events well and performs well on floods that are different in magnitude. It was also noticed that the ANN-6 model tolerates noisy inputs more than the MISOLM. Figure 4.45 shows that MISOLM can not simulate the rising and recession limb of the last two flood events very well. The MISOLM simulated hydrograph deviates from observed one in a bigger range than ANN-6 model. Although the ANN-6 model simulated hydrograph shows certain fluctuation, the comparison of performance between ANN-6 and MISOLM indicates that the ANN-6 model is insensitive to noisy data.

Table 4.8 The calibration and testing results of the MISOLM and ANN-6 models

Model	Calibration /Testing	Mean of obs.(cms)	RMSE (cms)	CE	R	MAE (cms)	Volume error (%)
MISOLM	Calibration	124	25.9	0.934	0.967	15.9	2.57
MISOLM	Testing	125	27.4	0.897	0.950	17.8	3.18
ANN-6	Testing	125	18.8	0.951	0.977	11.9	1.89

Note: both models are 6-hour lead-time model.

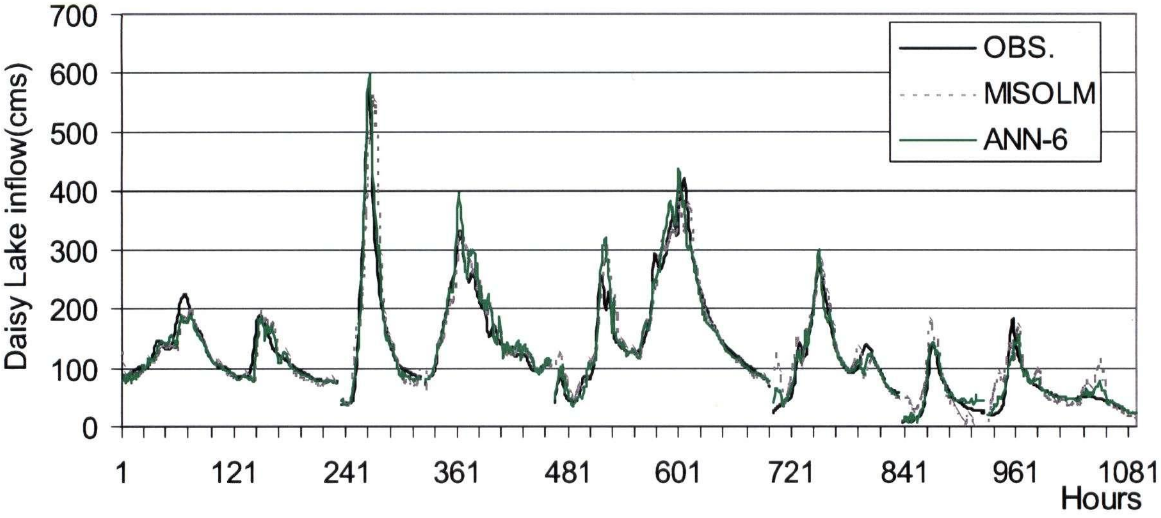

Figure 4.45 ANN-6 and MISOLM performance comparison on testing flood events

Table 4.9 lists the peak simulation results of the ANN-6 and the MISOLM. The two models simulated 8 out of 9 peaks with percentage error less than or equal to 20%. The ANN-6 simulated 7 out of 9 peaks with percentage error less than 10%. The ANN-6 simulated the peak of the biggest flood (No.3) much better than MISOLM. The comparison of peak simulation indicated that the ANN-6 is better than MISOLM although MISOLM is not bad.

Table 4.9 The peak simulation results of the MISOLM and ANN-6 models

Peak	Observation	ANN-6		MISOLM	
No.	Q_{peak}(cms)	Q_{peak}(cms)	Error (%)	Q_{peak} (cms)	Error(%)
1	225.7	180.5	-20.0	182.6	-19.1
2	191.0	180.0	-5.8	194.0	1.6
3	578.4	589.5	1.9	499.9	-13.6
4	326.2	357.3	9.5	315.6	-3.3
5	257.4	279.6	8.7	255.4	-0.8
6	418.6	383.1	-8.5	335.9	-19.8
7	269.6	295.4	9.6	285.6	5.9
8	143.0	142.6	-0.3	154.0	7.7
9	185.2	112.7	-39.1	113.0	-39.0

Note: Peaks in this table correspond to Figure 4.45

In conclusion, the hourly time step ANN models had good performance on either training or testing datasets, on either high flows or low flows; their performance on peak flow simulation was also very good. The performance comparison between the ANN-6 and the MISOLM showed that ANN-6 was better than MISOLM.

CHAPTER 5 Conclusion and Recommendation

ANNs have a dynamic and flexible architecture and simple computing elements that can be used to solve practical problems with natural mechanism. This research had successfully applied a three-layer feedforward ANN architecture, with an error backpropagation training algorithm and an early stop training technique, to build ANN models to forecast daily and hourly reservoir inflows. The results, from the ANN training, validation and real-time testing, indicate that the ANN is a promising tool for both real-time daily streamflow and hourly flash flood forecasting. Compared to sophisticated conceptual hydrological models, such as the UBCWM, the principal advantages of the ANN technique are that the model can be built much more easily and it is much faster and less cumbersome to calibrate than a conceptual model. Furthermore, the ANN9-5-1 model was shown to perform better than the UBCWM in both streamflow simulation and in real-time forecasting. The comparative results obtained from the ANN-6 model and MISOLM provide evidence that (1) the ANN model can provide accurate representation of the nonlinear and dynamic features of the modelled system; (2) the ANN model tolerates noisy data and have higher degree of reliability than other models; and (3) the ANN model can offer higher modelling and forecasting accuracy than MISOLM.

When an ANN model is used for streamflow forecasting, this research found that there are several issues that are very important to consider. Firstly, adequate knowledge of the physical processes involved (the rainfall-runoff process in this case) is important for ANN model inputs selection. It is obvious that ANN model inputs, which have real physical relationships with the output, will facilitate ANN model training and improve its performance. Secondly, having a sufficient amount of training data and appropriate partition of data into a training, monitoring and validation datasets are important for the ANN to give reliable results. The datasets should cover the full range of the rainfall-runoff mapping space. Otherwise, the ANN model will focus on the processes that dominate the training dataset. Appropriate division and clustering of the dataset is a technical maneuver to insure that hourly ANN models have similar performance on the training, monitoring and validation datasets, and have biggest generalization ability for future data. Finally, the LMBP combined with the early stop technique was found to perform best for the feedforward ANN training. Training the ANN with different initial weights and biases was found to be a good practice in helping to achieve better results, although it does not guarantee a global optimal solution. Compared with the LMBP, it was found that Genetic Algorithms (GA) were not a practical method for ANN training, although GA could theoretically reach a global optimal solution. The attempt at coupling GA with a mathematically rigorous ANN training method to train ANN was not successful in this research.

When using the ANN model developed in this research, water storage trend, *S,* need to be calculated continuously. For real-time hourly streamflow forecasting, *S* at the end of a day (or at the beginning of a day) should be calculated on a daily time step by daily data to take advantage of the relatively high quality daily data. At other times within a day, *S* could be calculated by modifying *S* at the beginning of a day. This method has the least chance to accumulate errors in *S* calculation.

Quality control of all ANN input information was found to be very important. Although an ANN model is more tolerant of noisy inputs than other models, it is always advisable to make all input information as accurate as possible. The hourly models developed did not use most recent streamflow (Q_t) as input, but they used Q_{t-1} as input. This deliberately selected streamflow input information will provide the user with one more hour to quality control past streamflow data during real-time flood forecasting.

In this research, the ANN models were calibrated and tested by following a strict procedure followed by hydrologists. It is expected that the performance of the ANN models with future data, should be similar to that obtained with historical data, provided that the input information is adequately long and of good quality, and provided that the rainfall-runoff mechanism does not change as a result of human activities or climate change. The ANN models need to be tested using real-time data. Whenever new major flood data are available, the ANN model should be retrained to accommodate the new information. The current hourly time step ANN models are not necessarily good for pure snowmelt streamflow forecasting because the models were trained mainly on flashy flood events.

It is likely that the ANN models will be used in other watersheds in British Columbia. The single most important issue in adapting an ANN to a new watershed is that the ANN should be trained using a dataset for that particular new watershed. Other issues, such as model input selection, dataset partition, training and testing can be done by the procedures proposed and outlined in this thesis.

Historical streamflow, precipitation and forecasted future precipitation are good indicators for rainfall-runoff simulation, while past temperature, forecasted future temperature and computed watershed water storage trend are good indicators for snowmelt affected runoff simulation. It is not recommended that snow pack information and snow water equivalent be included as input indicators to the ANN models due to data quality, data availability, and the uncertainty caused by the additional modelling needed to calculate the snow-water equivalent.

Reference

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000. Artificial neural networks in hydrology. I: preliminary concepts. Journal of hydrologic Engineering, Vol.5(2), pp.115-123.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000. Artificial neural networks in hydrology. II: hydrologic application. Journal of hydrologic Engineering, Vol.5(2), pp.124-137.

Beven, K.J., 2000. Rainfall-runoff Modelling: the Primer. John Wiley & Sons Ltd, Baffins Lane, Chichester, West Sussex, PO19 1UD, England.

Burnash, R.J.C., 1995. The NWS River Forecast System- Catchment modelling. In Computer models in watershed hydrology, V.P. Singh (Ed.) Water Resources Publications, pp.165-214.

Chang, F.J., Chang, L.C., Huang, H.L., 2002. Real-time recurrent learning neural network for streamflow forecasting. Hydrological Processes, Vol.16(13), pp. 2577-2588.

Chang, F.J., Chen, Y.C., 2001. A counterpropagation fuzzy-neural network modelling approach to real time streamflow prediction. Journal of Hydrology, Vol.245, pp. 153-164.

Chang, L.C., Chang, F.J., Chiang, Y.M, 2004. A two-step-ahead recurrent neural network for streamflow forecasting. Hydrological Processes, Vol.18(1), pp.81-92.

Chiang, Y.M., Chang, L.C., Chang, F.J., 2004. Comparison of static-feedforward and dynamic-feedback neural networks for rainfall–runoff modelling. Journal of Hydrology, Vol.290, pp.297-311.

Coulibaly, P., Anctil,P., Bobée, B., 2000. Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. Journal of Hydrology, Vol.230, pp.244–257.

Demuth, H., Beale, M., 2004. Neural Network Toolbox-for use with MATLAB, the MathWorks, Inc.

Fausett L. V., 1994. Fundamentals of neural networks: architecture, algorithm and application. Prentice-Hall. ISBN: 0-13-334186-0.

Gautam, M.R., Watanabe, K., Saegusa, H., 2000. Runoff analysis in humid forest catchment with artificial neural network. Journal of Hydrology, Vol.235, pp. 117-136.

Gupta, H.V., Ksu, K., and Sorooshian, S., 1997. Superior training of artificial neural networks using weight-space partitioning. *Proc., IEEE Int. Conf. on Neural Networks*, Institute of Electrical and Electronics Engineers, New York.

Haykin, S., 1994. *Neural networks: a comprehensive foundation.* MacMillan, New York.

Imrie, C.E., Durucan, S., Korre, A., 2000. River flow prediction using artificial neural networks: generalization beyond the calibration range. Journal of Hydrology, Vol.233, pp. 138-153.

Institute of Hydrology Modelling Group, 1999. Survey of modelling approaches suitable for AIMWATER (European research project), Institute of Hydrology, Wallingford, Oxfordshire, UK.

Jain, A., Sudheer, K.P., Srinivasulu, S., 2004. Identification of physical processes inherent in artificial neural network rainfall runoff models. Hydrological Processes, Vol.18(3), pp. 571-581.

Jones, J.A.A., 1997. Global hydrology: processes, resources, and environment management. Addison Wesley Longman Limited, England UK.

Kachroo, R.K., Liang, G.C., 1992. River flow forecasting. Part 2. Algebraic development of linear modelling techniques. Journal of Hydrology, Vol.133. pp.17-40.

Kim, G., Barros, A.P., 2001. Quantitative flood forecasting using multisensor data and neural networks. Journal of Hydrology, Vol. 246, pp. 45-62.

Laio, F., Porporato, A., Revelli, R., Ridolfi, L., 2003. A comparison of nonlinear flood forecasting methods. Water Resources Research, Vol.39(5), 1129. pp.TNN21-TNN24.

Lin, G.F., Chen, L.H., 2004. A non-linear rainfall-runoff model using radial basis function network. Journal of Hydrology, Vol. 289, pp.1–8.

Maier, H.R., Dandy, G.C., 1996. The use of artificial neural networks for the prediction of water quality parameter. Water Resources Research, Vol.32(4), pp.1013-1022.

Moradkhani, H., Hsu, K.L., Gupta, H.V., Sorooshian, S., 2004. Improved streamflow forecasting using self-organizing radial basis function artificial neural networks. Journal of Hydrology, Vol. 295, pp.246–262.

Nash, J.E., Barsi, B.I., 1983. A hybrid model for flow processing on large catchments. Journal of Hydrology, Vol.65, pp.125-137.

Nash, J.E., Foley, J.J., 1982. Linear models of rainfall-runoff system. In Singh V.P. (Ed.) Rainfall-runoff relationships, problems of the international symposium on rainfall-runoff modelling. May 1981, Mississippi State University, USA, Water Resources Publication, pp51-66.

Nayak, P.C., Sudheer, K.P., Rangan, D.M., Ramasastri, K.S., 2004. A neuro-fuzzy computing technique for modelling hydrological time series. Journal of Hydrology, Vol.291, pp.52-66.

Pan, T.Y., Wang, R.Y, 2004. State space neural networks for short term rainfall-runoff forecasting. Journal of Hydrology, Vol.297, pp.34-50.

Quick, M.C., 1995. The UBC Watershed Model. In Computer models in watershed hydrology, V.P. Singh (Ed.) Water Resources Publications, pp.233-280.

Rajurkar, M.P., Kothyari, U.C., Chaube, U.C., 2004. Modelling of the daily rainfall-runoff relationship with artificial neural network. Journal of Hydrology, Vol. 285, pp. 96-113.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J.,1986. Learning internal representations by error propagation. Parallel distributed processing, vol.1, MIT Press, Cambridge, Mass., pp.318-362.

Sajikumar, N., Thandaveswara, B.S., 1999. A non-linear rainfall–runoff model using an artificial neural network. Journal of Hydrology, Vol. 216, pp.32–55.

Shamseldin, A.Y., 1997. Application of a neural network technique to rainfall-runoff modelling. Journal of Hydrology, Vol.199, pp. 272-294.

Shamseldin, A.Y., O'connor, K.M., 1996. A nearest neighbours linear perturbation model for river flow forecasting. Journal of Hydrology, Vol.179, pp.353-375.

Sivapalan, M., Bloschl, G., Zhang, L., Vertessy, R. 2003. Downward approach to hydrological prediction. Hydrological Processes, Vol.17, pp.2101-2111.

Smith, J., and Eli, R. N. (1995). Neural-network models of rainfall-runoff process. *J. Water Resour. Plng. and Mgmt.*, ASCE, Vol.121(6), pp.499–508.

Sudheer, K.P., Nayak, P.C., Ramasastri, K.S., 2003. Improving peak flow estimates in artificial neural network river flow models. Hydrological Processes, Vol.17(3), pp. 677-686.

Sugawara, M., 1995.Tank model. In Computer models in watershed hydrology, V.P. Singh (Ed.) Water Resources Publications, pp.165-214.

Tingsanchali, T., Gautam, M.R., 2000. Application of tank, NAM, ARMA and neural network models to flood forecasting. Hydrological Processes, Vol.14, pp.2473-2487.

Tokar, A.S., and Johnson, P.A., 1999. Rainfall-runoff modeling using artificial neural networks. *J. Hydrologic Engrg.*, ASCE, Vol.4(3), pp.232–239.

Xu, Z.X., Li, J.Y., 2002. Short-term inflow forecasting using an artificial neural network model. Hydrological Processes, Vol.16(12), pp.2423–2439.

Young, P.C., 2002. Advances in real-time flood forecasting. Phil. Trans. R. Soc. Lond. A(2002)360, pp.1433-1450.

Zealand, C.M., Burn, D.H., Simonovic, S.P., 1999. Short term streamflow forecasting using artificial neural networks. Journal of Hydrology, Vol. 214, pp.32–48.

Zhang, B., Govindaraju, R.S., 2003. Geomorphology-based artificial neural networks (GANNs) for estimation of direct runoff over watersheds. Journal of Hydrology, Vol. 273, pp. 18-34.

Zhao, R.J., 1984. Watershed hydrological modelling – Xinanjiang model and Shanbei model. WaterPower Press, Beijing China.

APPENDIX A: An Introduction to MISOLM

Kachroo and Liang (1992) introduced a Simple-Input Linear Model (SLM) as a naive model against which the performance of more sophisticated rainfall-runoff model. SLM postulates a linear time invariant relationship between rainfall p_t and discharge q_t at the watershed outlet. SLM simulates unit hydrograph (UH) in rainfall-runoff mechanism. The rainfall-runoff relationship can be expressed as:

$$q_t = G \sum_{j=t-m+1}^{t} p_j h_{t-j+1} + e_t \qquad \text{(A-1)}$$

where G is the gain factor (i.e. amplification) of the system, e_t is a error term, m is the memory/UH length of the system and h_i is a set of discrete pulse response ordinates such that:

$$\sum_{i=1}^{m} h_i = 1 \qquad \text{(A-2)}$$

Combining G and h_i together and supposing the number of observations is n, equation (A-1) can be rewritten in matrix form:

$$\begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_m \\ q_{m+1} \\ \vdots \\ q_n \end{bmatrix} = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ p_2 & p_1 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ p_m & p_{m-1} & \cdots & p_1 \\ p_{m+1} & p_m & \cdots & p_2 \\ \vdots & \vdots & \cdots & \vdots \\ p_n & p_{n-1} & \cdots & p_{n-m+1} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_m \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \\ e_{m+1} \\ \vdots \\ e_n \end{bmatrix} \qquad \text{(A-3)}$$

or vector form:

$$\mathbf{Q} = \mathbf{P}\hat{\mathbf{H}} + \mathbf{E} \qquad \text{(A-4)}$$

The least squared error solution for $\hat{\mathbf{H}}$ is:

$$\hat{\mathbf{H}} = \left(\mathbf{P}^{\mathrm{T}}\mathbf{P}\right)^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{Q} \qquad \text{(A-5)}$$

Multivariate Regression (MR) is a simple mathematical method that can be used to simulate the relationship between system inputs and output. The linear regression takes the form of:

$$q_t = \sum_{i=1}^{n} h^i X_t^i \qquad \text{(A-6)}$$

where q_t is the system output time series (at time *t*), X_t^i is the i^{th} input time series, h^i is the coefficient for the i^{th} input. (A-6) can be written in vector form:

$$\mathbf{Q} = h^1\mathbf{X}^1 + \cdots + h^n\mathbf{X}^{\mathbf{n}} \qquad \text{(A-7)}$$

where $\mathbf{Q} = [q_1 \quad \cdots \quad q_t]^T$, $\mathbf{X}^{\mathbf{n}} = [X_1^n \quad \cdots \quad X_t^n]^T$.

The Multiple-input Single-output Model (MISOLM) is a mixture of SLM and MR model. It combines (A-4) and (A-7) together and takes the form of:

$$\mathbf{Q} = \mathbf{P}\hat{\mathbf{H}} + h^1\mathbf{X}^1 + \cdots + h^n\mathbf{X}^{\mathbf{n}} + \mathbf{E} \qquad \text{(A-8)}$$

or

$$\mathbf{Q} = \begin{bmatrix}\mathbf{P} & \mathbf{X}^1 & \cdots & \mathbf{X}^{\mathbf{n}}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{H}} \\ h^1 \\ \vdots \\ h^n\end{bmatrix} + \mathbf{E} \qquad \text{(A-9)}$$

$\begin{bmatrix}\hat{\mathbf{H}} & h^1 & \cdots & h^n\end{bmatrix}^T$ can be easily solved by least squares method with a similar form as (A-5). MISOLM can takes rainfall and other information related to runoff as model input.

APPENDIX B: Brief Introduction to UBC Watershed Model

The UBC Watershed Model (UBCWM) was mainly developed by Dr. M.C. Quick of University of British Columbia. The model is designed primarily for mountainous watersheds and calculates the total contribution from both snowmelt and rainfall runoff. A separate calculation can also be made of runoff occurring from glacier covered areas. The model is widely used in British Columbia and other areas, including the Himalayas in India and Pakistan. A detailed description of UBCWM can be found in Quick (1995). The following is a brief description of the UBCWM by Quick (1995).

B.1 An Overview of UBCWM

UBCWM, as a streamflow modelling system, has seven major components which form a logical subdivision of the hydro-meteorological modelling and evaluation process, and these components are:

(1) The meteorological sub-model, which distributes the input data to all elevation zones of the watershed. This distribution process is the most important part of the total modelling process because (a) it controls the total volume of moisture which is input to the model, and (b) it specifies the variation of temperature with elevation, which controls whether precipitation falls as rain or snow and also controls the melting of the snowpacks and glaciers.

(2) The soil moisture sub-model controls the evaporation losses and the subdivision of the rainfall and snowmelt into the four components of runoff, which are fast, medium, slow and very slow components. The components can be thought of as surface, or near surface runoff, interflow, and upper and lower groundwater runoff. The model computes the soil moisture deficit and it is this deficit which controls the non-linear subdivision of rain and melt into the runoff components. The non-linear hydrologic response of the watershed is therefore determined by this soil moisture budgeting. There is also a flash runoff mechanism which operates when rainfall exceeds a certain threshold, and this produces increased fast runoff

response.

(3) The watershed routing sub-model determines the time distribution of runoff. Each of the four components of runoff determined by the soil moisture sub-model are subjected to storage routing using either cascades or single linear reservoirs. Because these reservoirs are linear, conservation of mass is guaranteed and an accurate water budget is maintained. It has been found that this linear routing works well because the non-linear watershed response is handled by the soil moisture sub-model. Typically the time constants for this time distribution are of the order of one day or less for the fast component, five to ten days for the medium component, thirty to fifty days for the slow, upper groundwater component and one hundred to two hundred days for the very slow, lower groundwater component. It is because these time constraints are so distinctly different that it is possible to carry out a successful calibration of these components.

(4) The output and evaluation sub-model is designed to give flexible access to many aspects of the calculated watershed behavior. This information can be examined either in numerical format or graphically and, in addition, the data can be analyzed statistically. Part of this statistical output is used by the calibration sub-model.

(5) The semi-automatic calibration sub-model requires some user guidance to ensure that parameters are restricted to reasonable ranges. The calibration process is a constrained iterative search optimization which evaluates a maximum of four parameters at a time. Each of the three primary sub-models, meteorological, soil- moisture and time distribution, are sequentially optimized within the constraints set by the operators.

(6) The updating sub-model is based on a combination of feedback information from flow measurement and snow cover data from snow course or satellite. Both error sources must indicate a similar correction requirement before an adjustment is made.

(7) The routing sub-model, based on the UBC Flow Model, combines watershed flows and routes these flows through a river, lake and reservoir system.

B.2 Summary of UBCWM Structure

The UBCWM is designed to operate on a variable time step which can range from one hour to 24 hours, depending on the availability of the meteorological data inputs of maximum and minimum temperatures and precipitation. From these inputs the model calculates estimates of daily watershed outflows. Additional information is available on the accumulation and depletion of snowpack, the soil moisture status, the various soil and groundwater storage values and information on the contributions to runoff from various portions of the watershed and various surface and sub-surface components of runoff.

The basic structure of the model depends on a division of the watershed into a number of elevation bands. The elevation increment and the area for each band must be specified. The model has been used for watersheds ranging from a few square kilometers up to areas of several thousand square kilometers. The factors influencing choice of watershed size are the available streamflow reference data for calibration and the available meteorological data base.

In general, the meteorological data base is sparse for most of the mountainous regions modelled. In the majority of situations the meteorological data is from valley stations. As a result of these data constraints, an important aspect of the UBCWM is the elevation distribution of data. Functional relationships are specified describing the variability of temperature lapse rates. The temperature lapse rate is a key relationship because it influences the precipitation distribution, and also is very significant in determining snowmelt rates at various elevations. Precipitation inputs are made functionally dependent on elevation and on temperature regime. This functional variation of precipitation automatically recognizes that precipitation undergoes greater grographic enhancement in winter than it does during warm summer rainstorms. The importance of these temperature and precipitation gradients is illustrated in the examples of watershed calibration. The general structure of the UBCWM is indicated in Figure B1.

The response of the watershed to snowmelt and rainfall is controlled by a soil moisture model. The soil moisture status of each area-elevation band control the subdivision of the total snowmelt and rain input into the various components of watershed runoff response. These components of runoff can be characterized as fast, medium, slow and very slow runoff, and they may be conceptually thought of as representing surface runoff, interflow and superficial and deep groundwater components. The total snowmelt and rain input to each watershed band is subdivision on a priority basis, for example, a first priority is the satisfying of and soil moisture deficit which arises continuously because of evaporative demand.

Each component of runoff undergoes delay before reaching the outflow point of the watershed. These delays, or time distribution runoff, are achieved by using unit hydrograph routing. The various delay processes, or time distribution processes, can be thought of in terms of cascades of linear reservoirs.

For detailed algorithms used to describe the processes involved in the runoff process, please refer to Quick (1995).

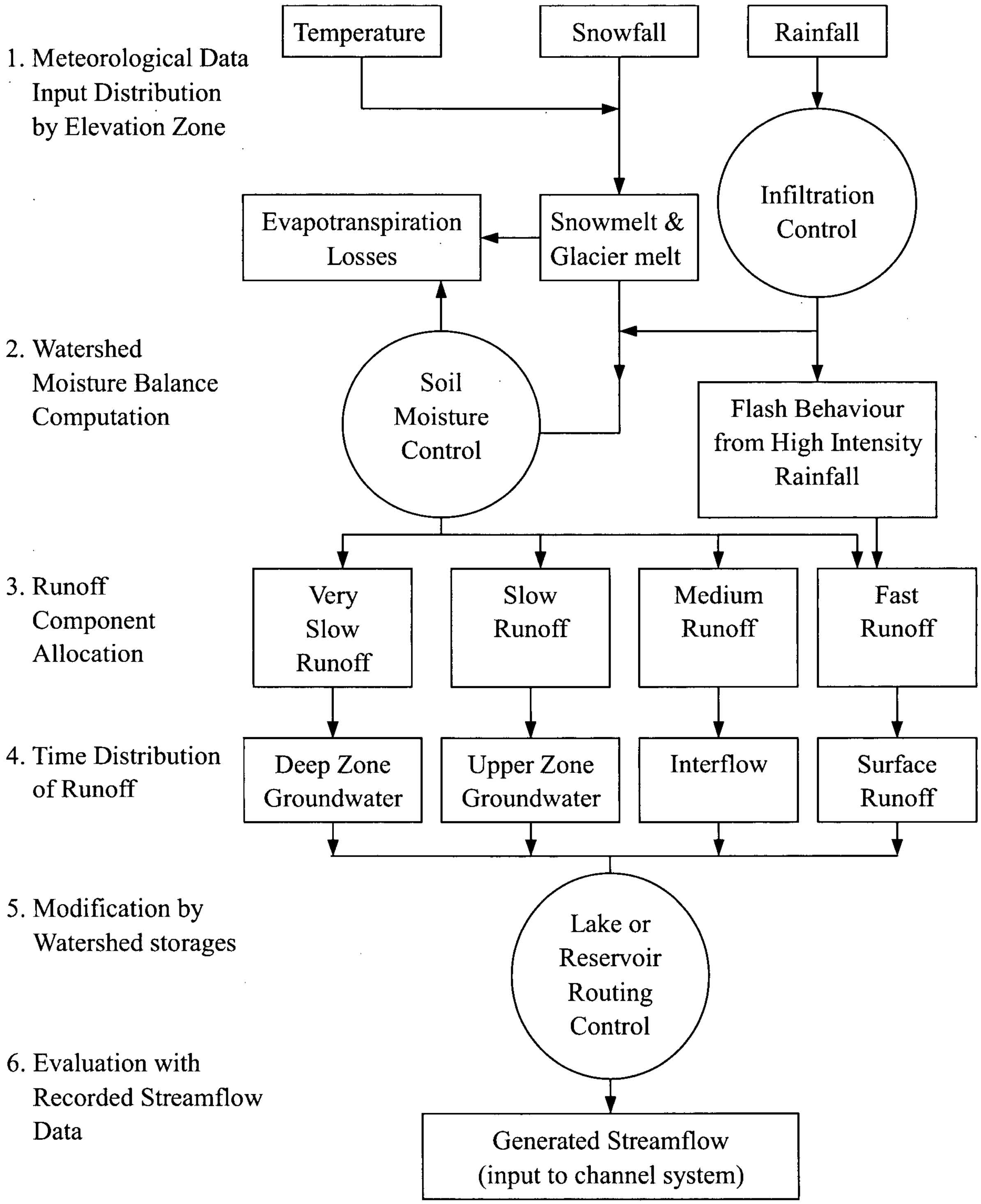
1. Meteorological Data Input Distribution by Elevation Zone
Temperature
Snowfall
Rainfall
Evapotranspiration Losses
Snowmelt & Glacier melt
Infiltration Control
2. Watershed Moisture Balance Computation
Soil Moisture Control
Flash Behaviour from High Intensity Rainfall
3. Runoff Component Allocation
Very Slow Runoff
Slow Runoff
Medium Runoff
Fast Runoff
4. Time Distribution of Runoff
Deep Zone Groundwater
Upper Zone Groundwater
Interflow
Surface Runoff
5. Modification by Watershed storages
Lake or Reservoir Routing Control
6. Evaluation with Recorded Streamflow Data
Generated Streamflow (input to channel system)

Figure B1 UBCWM general flow chart (Quick, 1995)

APPENDIX C: Source Code

C.1 Source Code for ANN Models Training

```
%Matlab program for ANN model training. Jian Li, September 2004
%The program uses Levenberg-Marquardt Back Propagation training method: trainlm
%and Early Stop technique. The program call a subroutine, outputffs, during training.
%Arguments of the program:
%nInput: number of ANN inputs, it is also the number of ANN input nodes
%nNodes: number of hidden nodes
%nIter: training times
%fn: file name for training results
%Default file needed for ANN training:
%File name of input pattern: input.txt
%File name of observed streamflow(desired ANN output): output.txt

function d=trainffs(nInput,nNodesH,nIter,fn)

global stat_indices weight_12 biasH weight_23 biasO maxI minI maxO minO outputlessthan;
global fmtHW fmtHB outputpath;

nRows=4434;     % number of historical data for 1h lead-time ANN model
outputpath=fn;
outputpath=[outputpath,'_'];
outputlessthan=12;
outputpoint=2;
ndatadiv=[2673 668 1093 4434];%1 hour ahead model, number of rows for calibration, verification and testing

stat_indices=zeros(nIter,13);
weight_12=zeros(nNodesH,nInput,nIter);
biasH=zeros(nNodesH,nIter);
weight_23=zeros(nIter,nNodesH);
biasO=zeros(nIter,1);
rangeI=zeros(nInput,2);
rangeI(:,2)=1;

fmtHW='';
```

```
for n=1:nInput
  fmtHW=[fmtHW,'%10.6f     '];
end
fmtHW=[fmtHW,'\r\n'];

fmtHB='';
for n=1:nNodesH
  fmtHB=[fmtHB,'%10.6f      '];
end
fmtHB=[fmtHB,'\r\n'];

%%%scale_min=0.15;scale_max=0.85
filename=outputpath;
filename=[filename,'output.txt'];
fid=fopen(filename,'r');
myOO=fscanf(fid,'%f',[1,nRows]);
status=fclose(fid);
%plot(1:nRows,myOO);

filename=outputpath;
filename=[filename,'input.txt'];
fid=fopen(filename,'r');
%%ndatadiv=fscanf(fid,'%d',[1,4]);
myII=fscanf(fid,'%f',[nInput,nRows]);
status=fclose(fid);
ndatadiv(1,2)=ndatadiv(1,1)+ndatadiv(1,2);
ndatadiv(1,3)=ndatadiv(1,2)+ndatadiv(1,3);
%plot(1:nRows,myII(1,:));

maxI=max(myII');
minI=min(myII');
maxO=max(myOO');
minO=min(myOO');
tmp_I=(maxI-minI)/1.4*0.3;
tmp_O=(maxO-minO)/1.4*0.3;
maxI=maxI + tmp_I;
minI=minI - tmp_I;
maxO=maxO + tmp_O;
minO=minO - tmp_O;

for n=1:nRows
  myO(n)=(myOO(n)-minO)/(maxO-minO); % scale to 0.15 to 0.85
  for m=1:nInput % modify here
    myI(m,n)=(myII(m,n)-minI(m))/(maxI(m)-minI(m));
  end
end

myItr=myI(:,1:ndatadiv(1,1));myOtr=myO(1:ndatadiv(1,1)); % 6 year's data trainning
v.P=myI(:,ndatadiv(1,1)+1:ndatadiv(1,2));v.T=myO(ndatadiv(1,1)+1:ndatadiv(1,2));% 2 year's data for vailidation
t.P=myI(:,ndatadiv(1,2)+1:ndatadiv(1,3));t.V=myO(ndatadiv(1,2)+1:ndatadiv(1,3));% 4 year's data for testing
obsO=myOO(ndatadiv(1,2)+1:ndatadiv(1,3));

net=newff(rangeI,[nNodesH,1],{'logsig','purelin'},'trainlm');
net.layers{1}.initFcn='initnw';
net.inputWeights{1,1}.initFcn='initnw';
net.biases{1,1}.initFcn='rands';
net.biases{2,1}.initFcn='rands';
%% end of initialize.
%% begin the training loop

for itr = 1 : nIter
  itr
      net=init(net);
      %net.trainParam.show=10;
      net.trainParam.lr=0.1;
      net.trainParam.epochs=300;
      %net.trainParam.goal=0.05;

      net=train(net,myItr,myOtr,[],[],v,t);% Early stop
      %net=train(net,myItr,myOtr); % no early stop
      Y=sim(net,t.P);

      %=== restore to orginal value  ===
      calO=minO + Y.*(maxO-minO);

   if nIter==1
%       nn0=[0 87 135 231 135 231 87 159];%for 6 hour ahead model.
      %nn0=[0 88 136 232 136 232 88 160];%for 6 hour ahead model.
      %nn0=[0 241 97 145 241 145 97 169];%2 hour ahead org.
      %nn0=[0 234 90 138 234 138 90 162];%2 hour ahead -7.
      nn0=[0 235 91 139 235 139 91 163];%1 hour ahead -6.
```

```
        for n=3:8
            nn0(n)=nn0(n)+nn0(n-1);
        end
        for n=1:7
            figure(n);
            nn=nn0(n)+1:nn0(n+1);
            plot(nn,calO(nn),nn,obsO(nn));
            xlabel('timestep');
            ylabel('discharge(cms)');
            legend('ANN','Observed');
        end
        figure(8);
    end
       %=== Mean square error ===
       e=calO-obsO;
       MSE=mse(e);
   RMSE=sqrt(MSE);
   MAE=mae(e);
       %=== Model Coefficent of efficency ===
       mean_target=mean(obsO);% Mean of test value
       CE=1-MSE/mse(obsO-mean_target);
       [m,b,r]=postreg(calO,obsO);% post trainning regression
   % percentage errors
   percent_e=(e./obsO).*100; %%element in metrix e devided by coresponding element in metrix
calO

   nNeg=0;n20=0;
       [index1,index2]=size(calO);
       for n=1:index2
      if(calO(n)<0)
         nNeg=nNeg+1;
      end
      if(abs(percent_e(n))<=20)
         n20=n20+1;
      end
       end

   maxE=max(e');
   minE=min(e');
   Vol_E=(mean(calO)-mean_target)/mean_target*100;
   maxpercentE=max(percent_e');
   minpercentE=min(percent_e');
```

```
  %%store the parameters

stat_indices(itr,:)=[RMSE,CE,m,b,r,MAE,nNeg,n20,maxE,minE,maxpercentE,minpercentE,Vol_
E];
  weight_12(:,:,itr)=net.IW{1,1};
  biasH(:,itr)=net.b{1};
  weight_23(itr,:)=net.LW{2,1};
  biasO(itr)=net.b{2};

 if mod(itr,outputpoint)==0
    outputffs(nInput,nNodesH,itr);
 end

end
%%end of bath training

%data Output
filename=outputpath;
filename=[filename,int2str(nInput)];
filename=[filename,int2str(nNodesH)];
filename=[filename,'1_'];
filename=[filename,int2str(nIter)];
filename=[filename,'.txt'];

fid=fopen(filename,'w');
fprintf(fid,'ANN=%1d%1d1\r\n',nInput,nNodesH);
fprintf(fid,'No    RMSE  CE m   b    r    MAE    Negn20
     maxE\tminE\tmaxPE\tminPE\tVolE\r\n');
for n=1:nIter
     fprintf(fid,'%3d  %7.3f   %6.3f   %6.4f   %5.2f   %5.3f   %5.2f   %2d     %3d
     %5.1f\t%6.1f\t%6.2f %6.2f    %7.2f\r\n',n,stat_indices(n,:));
end

for n=1:nIter
  if stat_indices(n,1)<outputlessthan
     fprintf(fid,'\r\n No.%3d Weights IW\r\n',n);
       fprintf(fid,fmtHW,(weight_12(:,:,n))');
     fprintf(fid,'Bias for hidden layer\r\n');
       fprintf(fid,fmtHB,biasH(:,n));
     fprintf(fid,'Weights LW\r\n');
       fprintf(fid,fmtHB,weight_23(n,:)');
    fprintf(fid,'Bias for output layer\r\n');
   fprintf(fid,'%8.4f\r\n',biasO(n));
  end;
end
d=1;
%End of main program for ANN training


function outputffs(nInput,nNodesH,nIter)

global stat_indices weight_12 biasH weight_23 biasO maxI minI maxO minO outputlessthan ;
global fmtHW fmtHB outputpath;

%data Output
filename=outputpath;
filename=[filename,int2str(nInput)];
filename=[filename,int2str(nNodesH)];
filename=[filename,'1_'];
filename=[filename,int2str(nIter)];
filename=[filename,'.txt'];

fid=fopen(filename,'w');
fprintf(fid,'ANN=%1d%1d1\r\n',nInput,nNodesH);
fprintf(fid,'No    RMSE   CE  m    b    r     MAE     Negn20
    maxE\tminE\tmaxPE\tminPE\tVolE\r\n');
for n=1:nIter
    fprintf(fid,'%3d   %7.3f   %6.3f   %6.4f   %5.2f   %5.3f   %5.2f   %2d     %3d
    %5.1f\t%6.1f\t%6.2f %6.2f    %7.2f\r\n',n,stat_indices(n,:));
end

for n=1:nIter
  if stat_indices(n,1)<outputlessthan
    fprintf(fid,'\r\n No.%3d Weights IW\r\n',n);
      fprintf(fid,fmtHW,(weight_12(:,:,n))');
    fprintf(fid,'Bias for hidden layer\r\n');
      fprintf(fid,fmtHB,biasH(:,n));
    fprintf(fid,'Weights LW\r\n');
      fprintf(fid,fmtHB,weight_23(n,:)');
    fprintf(fid,'Bias for output layer\r\n');
   fprintf(fid,'%8.4f\r\n',biasO(n));
  end;
end
```

```
fprintf(fid,'\r\npremnmx parameters. The min and max for each input is:\r\n');
fprintf(fid,'%8.2f',minI);fprintf(fid,' <<=min_for_input\r\n');
fprintf(fid,'%8.2f',maxI);fprintf(fid,' <<=max_for_input\r\n');
fprintf(fid,'%8.2f.........................................%8.2f <<=min_and_max_for_output',minO,maxO);
fclose(fid);
%End of subroutine outputffs
```

C.2 Source Code for MISOLM Model Calibration

```
use imslf90
use lin_sol_gen_int
use rand_gen_int
use error_option_packet

implicit none

double precision, allocatable :: A(:,:),X(:),Y(:),RES(:)
double precision, allocatable :: PQ(:,:),Cal(:)
CHARACTER*30, iName(10),oNameOBS, oNameUH, oNameData
CHARACTER*100, StaName
REAL Area
Integer nYear, nCal, yy,mm,dd,hh,dt
integer nFile,nTimestep
Integer nUH(10)

Integer n,i,j,k,uh,nBeg,iPflag,NCA,NRA,LDA
CHARACTER*404, cTmp
Integer y0,m0,d0,h0,dt0,nRec

!= = = = = = = = = = = Read general information data file: GIF.txt
Open (unit=1, FILE='GIF.txt',STATUS='old')
read (1,*)StaName     !description of the file
read (1,*)StaName     !Watershed name
read (1,*)Area        !Area
read (1,*)nFile, nCal,nYear,nTimestep  !number of model input, number of year of calibration
                                       !data, number of year of data, number of data in each year
read (1,*)yy,mm,dd,hh,dt
read (1,*)(nUH(n),n=1,nFile)  !memory/UH length
read (1,*)oNameOBS            !file name of observed streamflow
```

```
do 10 n=1,nFile                 !file name of each input
    read(1,*)iName(n)
10  continue
read (1,*)oNameUH               !UH file name
read (1,*)oNameData             !Output file name
close(1)
!= = = = = = = = = = = = = = = = = = = = = =end of reading GIF.txt
NRA=nCal*nTimestep
NCA=0
do 20 n=1,nFile
    NCA=NCA+nUH(n)
20  continue

Allocate
(A(NRA,NCA),X(NCA),Y(NRA),RES(NRA),PQ(nYear*nTimestep,nFile+1),Cal(nYear*nTimes
tep))
do 30 n=1,nFile
    write(*,*) ' Reading file:'//trim(iName(n))
    OPEN (UNIT=1, FILE=iName(n),STATUS='old')  ! open and read each input file
    read(1,*) cTmp   ! conmment
    read(1,*) ipflag   !sign for precipitation or not (0/other)
    read(1,*) y0,m0,d0,h0,dt0,nRec
    if(yy.ne.y0 .or. mm.ne.m0) then
        write (*,*)'begin time error when reading', iName(n)
    end if
    if(nRec .ne. nYear*nTimestep) then
        write(*,*)'Total time step error when reading',iName(n)
    end if
    if(iPflag.eq.0) then ! precipitation data
        do 40 i=1,nRec
            read (1,*)y0,m0,d0,h0,PQ(i,n)
40      continue
    else !other data
        do 50 i=1,nRec
            read (1,*)y0,y0,m0,d0,h0,PQ(i,n)
50      continue
    end if
    CLOSE(1)
30  continue  !end of reading input data

write(*,*) ' Reading file:'//trim(oNameOBS)          !Read observed streamflow data
OPEN (UNIT=1, FILE=oNameOBS,STATUS='old')
```

```
    read(1,*) cTmp   ! conmment
    read(1,*) y0  !sign for precipitation or not (0/other)
    read(1,*) y0,m0,d0,h0,dt0,nRec
    if(yy.ne.y0 .or. mm.ne.m0) then
        write (*,*)'begin time error when reading', oNameOBS
    end if
    if(nRec .ne. nYear*nTimestep) then
        write(*,*)'Total time step error when reading',oNameOBS
    end if
    do 60 i=1,nRec
60  read (1,*)y0,y0,m0,d0,h0,PQ(i,nFile+1)
close(1)
!= = = = = = = = = = = = = = = = = = = = = = end of reading input file
!All data is stored in to PQ(:,:), the last column is observed discharge
!the other columns are input date from each of the input file.

n=maxloc(nUH,1)
n=nUH(n)
do 70 i=1,n-1
    do  70 j=1,nFile
        A(i,j)=0.0
70  continue

j=1
do 80 n=1,nFile
    do 90 i=1,nUH(n)
        do 100 k=i,NRA
            A(k,j)=PQ(k-i+1,n)
100     continue
        j=j+1
90  continue
80  continue
!= = = = = = = = = = = = = = = = = = = = = = =end of generating A, the coefficient matrix.
! A consist of the data for calibration. each inputs was expanded to several column(s)
! according to the memory length.

do 110 n=1, NRA
    Y(n)=PQ(n,nFile+1)
110 continue
!= = = = = = = = = = = = = = = = = = = = = = =end of generating Y, for AX=Y, every thing is ready
!= = = = = = = = = = = = = = = = = = = = = = =to solve the equation:
Write(*,*) ' Begin to calibrate...'
```

```
LDA=NRA
CALL DLSQRR (NRA, NCA, A, LDA, Y, 0, X, RES, k)
Write(*,*) ' Finish calibrating.'
!= = = = = = = = = = = = = = = = = = = = = = end of solving AX=Y
!Residue RES=B-AX
!= = = = = = = = = = = = = = = = = = = = = =
n=maxloc(nUH,1)
n=nUH(n)
do 120 i=1,n-1
     Cal(i)=0.0
120 continue

do 130 n=n,nRec
     Cal(n)=0.0
     uh=1
     do 140 i=1,nFile
          nBeg=n
          do 150 j=1,nUH(i)
               Cal(n)=Cal(n)+PQ(nBeg,i)*X(uh)
               uh=uh+1
               nBeg=nBeg-1
150       continue
140 continue
130 continue
!= = = = = = = = = = = = = = = = = = = = = = end of calculating model output

write(*,*)' Output...'
open (unit=1, file=oNameUH,Status='unknown')
n=1
write (1,*)'Catchment: '//trim(StaName)
write (1,*)'The Calibrated UH for Each Inflow(input)'
write (1,*) nFile,' <=Number of input'
do 200 i=1,nFile
     write(1,*) trim(iName(i)),'      ', nUH(i),' <=memory length'
     do 210 j=1,nUH(i)
          write(1,*) X(n)
          n=n+1
210 continue
200 continue
close(1)

open (unit=1, file=oNameData,Status='unknown')
```

```
write(1,*)'OBC          Cal'
n=nFile+1
do 220 i=1,nRec
     write(1,'(1x,f10.3,1x, f10.3)')PQ(i,n),Cal(i)
220 continue
close(1)

!= = = = = = = = = = = = = = = = = = = = = = output results
cTmp='(1x'
do 310 n=1,NCA
     CTmp=trim(CTmp)//',1x,f7.2'
310 continue
cTmp=trim(cTmp)//')'

open (unit=1, file='A.txt', status='unknown')
do 320 n=1,NRA
     write (1,cTmp) (A(n,i),i=1,NCA)
320 continue
write (1,*)'Y RES'
do 330 n=1,NRA
     write (1,'(f8.2,1x,f8.3)') Y(n),RES(n)
330 continue
write (1,*)'X(i)='
do 340 n=1, NCA
     write(1,*) x(n)
340 continue
write (1,'("n=",i2)')k
close(1)

Deallocate (A,X,Y,RES,PQ,Cal)
write (*,*)' Finish.'
END
```