HYDROLOGIC PREDICTION USING PATTERN RECOGNITION AND SOFT-COMPUTING TECHNIQUES

A Thesis Submitted to the

College of Graduate Studies and Research

In Partial Fulfillment of the Requirements For the

Degree of Doctor of Philosophy

In the

Department of Civil and Geological Engineering

University of Saskatchewan

Saskatoon

By

Kamban Parasuraman

Permission to use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by Dr. Amin Elshorbagy who supervised my thesis work or, in his absence, by the Head of the Civil and Geological Engineering or the Dean of the College of Engineering. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Civil and Geological Engineering

University of Saskatchewan

Saskatoon, Saskatchewan

Canada, S7N 5A9

Abstract

Several studies indicate that the data-driven models have proven to be potentially useful tools in hydrological modeling. Nevertheless, it is a common perception among researchers and practitioners that the usefulness of the system theoretic models is limited to forecast applications, and they cannot be used as a tool for scientific investigations. Also, the system-theoretic models are believed to be less reliable as they characterize the hydrological processes by learning the input-output patterns embedded in the dataset and not based on strong physical understanding of the system. It is imperative that the above concerns needs to be addressed before the data-driven models can gain wider acceptability by researchers and practitioners.

In this research different methods and tools that can be adopted to promote transparency in the data-driven models are probed with the objective of extending the usefulness of data-driven models beyond forecast applications as a tools for scientific investigations, by providing additional insights into the underlying input-output patterns based on which the data-driven models arrive at a decision. In this regard, the utility of self-organizing networks (competitive learning and self-organizing maps) in learning the patterns in the input space is evaluated by developing a novel neural network model called the spiking modular neural networks (SMNNs). The performance of the SMNNs is evaluated based on its ability to characterize streamflows and actual evapotranspiration process. Also the utility of self-organizing algorithms, namely genetic programming (GP), is evaluated with regards to its ability to promote transparency in data-driven

models. The robustness of the GP to evolve its own model structure with relevant parameters is illustrated by applying GP to characterize the actual-evapotranspiration process. The results from this research indicate that self-organization in learning, both in terms of self-organizing networks and self-organizing algorithms, could be adopted to promote transparency in data-driven models.

In pursuit of improving the reliability of the data-driven models, different methods for incorporating uncertainty estimates as part of the data-driven model building exercise is evaluated in this research. The local-scale models are shown to be more reliable than the global-scale models in characterizing the saturated hydraulic conductivity of soils. In addition, in this research, the importance of model structure uncertainty in geophysical modeling is emphasized by developing a framework to account for the model structure uncertainty in geophysical modeling. The contribution of the model structure uncertainty to the predictive uncertainty of the model is shown to be larger than the uncertainty associated with the model parameters. Also it has been demonstrated that increasing the model complexity may lead to a better fit of the function, but at the cost of an increasing level of uncertainty. It is recommended that the effect of model structure uncertainty should be considered for developing reliable hydrological models.

Acknowledgements

I would like to express my earnest gratitude and appreciation to my respected supervisor, Dr. Amin Elshorbagy, for his invaluable guidance, support and encouragement throughout my work during the Ph.D. program. His critical appraisal and suggestions have been priceless at every stage of this research.

I would like to extend my acknowledgement to my advisory committee members Dr. S. L. Barbour, Dr. B. C. Si, and Dr. C. Zhang for their valuable suggestions and feedback. The time and support of my advisory committee chair, Dr. C. D. Hawkes is greatly acknowledged. I would also like to acknowledge, Dr. S. K. Carey, Carleton University, for his contributions to this work.

I take this opportunity to thank my colleagues at Centre for Advanced Numerical Simulation (CANSIM), my friends, and staff at the Department of Civil and Geological Engineering, for making my stay at the University of Saskatchewan a memorable one.

Furthermore, I would like to thank Dr. Amin Elshorbagy, and the Department of Civil and Geological Engineering, for providing the financial support for this research work.

Last but not the least; I am extremely grateful to my parents and sisters for their unwavering guidance, support, and motivation, throughout my Ph.D. program.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

2D	2-Dimensional
3D	3-Dimensional
ANNs	Artificial neural networks
ARMA	Autoregressive moving average
ARMAX	Autoregressive moving average with exogenous inputs
AT	Air temperature
BD	Bulk density
BP	Back-propagation
BR	Bayesian regularization
CV	Coefficient of variation
DA	Decision analysis
EA	Evolutionary algorithm
EBBR	Energy balance Bowen ratio
EC	Eddy-covariance
ET	Evapotranspiration
FAO	Food and Agriculture Organization
FF-NN	Feed-forward neural networks
G	Ground heat flux
GAs	Genetic algorithms
GP	Genetic programming
GSR	Genetic symbolic regression
GT	Ground temperature
H	Sensible heat flux
LE	Latent heat flux
MCDA	Multi-criterion decision analysis
MISO	Multiple-input-single-output
MR	Mean residual
MRE, MARE	Mean absolute relative error
MSE	Mean sum of squares of network errors
MSW	Mean of the sum of squares of the network weights and bias

NR	Net radiation
PM	Penman-Monteith
PTFs	Pedotransfer functions
R	Coefficient of correlation
RH	Relative humidity
RMSE	Root-mean squared error
SARIMA	Seasonal auto-regressive integrated moving average
SBH	South Bison hill
SD	Standard deviation
SISO	Single-input-single-output
SMNNs	Spiking-modular neural networks
SOLO	Self-organzing linear output
SOM	Self-organizing map
SONO	Self-organizing nonlinear output
SSC	Sand, silt, and clay content
SWSS	South-west sand storage
U	Uncertainty
UNSODA	UNsaturated SOil hydraulic DAtabase
WS	Wind speed

Chapter 1 Introduction

1.1 Background

Hydrology may be defined as the science that attempts to answer the question, "What happens to rain?" (Penman, 1961). In an attempt to address the above query, a plethora of studies have been carried out in the past to characterize the different components of the hydrological cycle including, but not limited to, precipitation, evaporation, infiltration, ground water flow, and runoff. Experience has shown that characterizing the above processes still remains a daunting task, as these processes are embedded with high nonlinearity in both spatial and temporal scales. Although it may not be possible to fully address the above intricate query, research efforts are being focused on explaining the hydrological processes based on extension of observation and theory by developing *models*. A *model* can be defined "as a simplified representation of the essential aspects of an existing system (or a system to be constructed), which presents the knowledge of the system in a usable form" (Eykhoff, 1974).

Hydrological models can be classified into different classes based on different criteria: (i) based on process description; hydrologic models can be classified as lumped or distributed, deterministic or stochastic or mixed, (ii) based on time scale; hydrologic models can be classified as event based, continuous time, and large time-scale, and (iii) based on solution technique; hydrologic models can be classified as numerical, analog, and analytical models (Singh, 1995).

In a broader perspective, hydrological models can be classified into a) conceptual or mechanistic models, and (b) black-box or data-driven models. The key differentiator between these two modeling types is that; mechanistic models give a physical insight of the system and can be built when the system is not yet constructed. Usually a set of differential equations supplemented with algebraic equations is used to give a mathematical description of the model. On the contrary, data-driven models attempt to develop relationships among the input and output variables involved in a physical process without considering a profound understanding of the underlying physical process. Construction of hydrological models can be accomplished by two approaches: (i) inductive approach, and (ii) deductive approach. In the case of inductive approach, the first step is to deduce a hypothesis based on a set of observations (inputs and outputs). The deduced hypothesis is then tested using a different set of observations, before being applied to a new set of observations. All data-driven models belong to the class of inductive approach. However, in the case of deductive approach, contrary to the inductive approach, a hypothesis is made straight away based on our knowledge of the system. The output from the system can then be calculated using the inputs and the assumed hypothesis. All conceptual models belong to the class of deductive approach.

Conceptual mechanistic models that attempt to capture the characteristics of the underlying physical process through the equations of mass, momentum, and energy, are the most widely adopted models in hydrological literature. These conceptual models have been developed based on certain baseline assumptions, at a particular scale of interest, for some measure of meteorological and topographical control over the hydrological

processes. Nevertheless, these models are being widely adopted at all spatial and temporal scales, and the above mentioned shortcomings are tackled implicitly by the complicated and ad-hoc calibration process. This highlights the fact that, for a hydrological process under consideration, the ability of a conceptual model in providing reasonable estimates depends upon the success of the adopted calibration scheme. However, calibration of a conceptual hydrological model is not straight-forward, and is prone to several difficulties, requiring sophisticated mathematical tools, a significant amount of calibration data, and some degree of expertise and experience with the model (Duan et al., 1992). As a result, system-theoretic models are starting to be widely acknowledged as suitable alternatives to model the complex hydrological processes due to their ability to learn the input-output dynamics in the data without having the complete physical understanding of the system. The success of the data-driven models in modeling hydrological processes can be attributed to their intrinsic generality, flexibility, and global performance in most applications where other models tend to fail or become cumbersome (Shamseldin et al., 2002).

1.2 Area of interest

Linear time series models (e.g. autoregressive moving average (ARMA); autoregressive moving average with exogenous inputs (ARMAX)) are the most traditionally adopted data-driven models for characterizing hydrological time series, because such models are accepted as a standard representation of a stochastic time series (Maier and Dandy, 1997). As the linear time series models make use of classical statistics to analyze the historical data, they do not attempt to represent the nonlinear dynamics, if

any, between the input and the output variables. Nevertheless, most hydrological processes exhibit high nonlinearity between the input and the output variables, and hence in such cases, the linear time series models may not always perform well (Hsu et al., 1995). In the past, owing to the difficulties associated with nonlinear model structure identification and parameter estimation, the usual practice was to assume linearity or piecewise linearity in modeling nonlinear hydrological processes (Hsu et al., 1995).

Over the past few decades, advancement in computer power and technology has provided significant impetus to the way data-driven models are built to characterize the hydrological processes. The traditional methods of estimating dependencies from hydrological data using statistics (multivariate regression and classification), were slowly replaced by new techniques, which were not often based on the assumptions of the well-behaved statistical distributions of random processes. One of the most exciting ideas that emerged from the vast pool of computer-based research is the thought of emulating the low-level mechanism of the human brain through artificial neural networks (ANNs). The idea of ANNs was first seeded by the pioneering work of McCulloch and Pitts (1943). However, the major developments behind the resurgence of ANNs occurred when the back-propagation algorithm for multilayer perceptrons (multilayer feed forward networks) was first proposed by Werbos (1974), and then reinvented several times before being popularized by Rumelhart et al. (1986). Similarly, another important technique to have emerged in the last decade, and which is being widely acknowledged as an important tool in the inventory of machine learning methods is the Genetic Programming

(GP). GP is an evolutionary algorithm that is based on the concepts of natural selection and genetics, and was first proposed by Koza (1992).

1.3 Problem recognition

The emergence of the ANNs, on the positive side, has provided many promising results in the field of hydrology and water resources engineering (ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000a, 2000b; Maier and Dandy, 2000), leading to the creation of a new chapter in hydrology that has been termed "neurohydrology." However, on the darker side, until recently, ANNs were not readily accepted as a modeling tool by the wider hydrological community and decision makers, based on the perception that the ANNs are pure black-box models and they do not consider the underlying physics. Nevertheless, recent studies by Wilby et al. (2003), Jain et al. (2004), and Sudheer and Jain (2004) have demonstrated that the ANNs are not pure black-box models and it is possible to extract some of the physics involved. Hence lately, more research has been directed to identifying the mechanism by which ANNs learn the hydrological patterns embedded in the input-output data. Contrary to the ANNs, the advancements in the application of GP in hydrological literature is still in its nascent stage, irrespective of the fact that both the GP and the ANNs can be seen as alternative techniques for the same task, like, e.g., classification and approximation problems. This immaturity stems partly from the fact that GP, in addition to being a relatively new technique, is also computationally intensive and hence is consistently marginalized by the hydrological modellers.

Although several studies indicate that the data-driven models have proven to be potentially useful tools in hydrological modeling, two of the main issues that needs to be further explored before these models gain wider acceptability by researchers and practitioners are: (i) bringing transparency (insights), even if only partially, on the basic process by which these data-driven models arrive at a decision, and thereby extending their usefulness beyond forecast applications as a tool for scientific investigations; and (ii) identifying the effective ways for performing uncertainty analysis in the data-driven models, which in-turn contributes to improving the reliability of such models. Therefore this research, driven by the motivation of improving the credibility of the data-driven models among researchers and practitioners, is carried out to identify some of the possible solutions to the above mentioned issues.

1.4 Research Objectives

In pursuit of improving the credibility of the data-driven models in hydrological modeling, the objectives of this research can be itemized as:

1. Extending the usefulness of the system-theoretic models beyond forecast applications as a tool for scientific investigations, by providing additional insights into the underlying input-output patterns on which the data-driven models arrive at a decision.
2. Improving the reliability of the system-theoretic models by identifying ways for incorporating uncertainty estimates as part of the data-driven model building exercise.

1.5 Scope of the research

The present work is part of a large research program that aims at developing a framework to help understand the hydrological processes that are dominant in reconstructed (reclaimed) watersheds, and thereby develop a sustainable reclamation strategy. A detailed overview of the research program can be found in (Boese, 2003), and for brevity, a concise description of the research program is given here. Large scale mining in the Athabasca basin, Alberta, Canada, involves stripping of large amount of organic and glacial deposits and a layer of saline/sodic cretaceous shale to gain access to the oil sands. Prior to mining, the shale overburden is stable since it is over consolidated, confined, and is exposed only to saline/sodic pore fluids. However, once the shale is placed on the surface, it is exposed to fresh water and oxygen and is susceptible to weathering, which in the long run affects the stability of the pile (Barbour et al., 2001). In order to overcome this problem, the piles are re-contoured and capped with sufficient soil cover so that the amount of precipitation percolating below the root zone can be minimized while maintaining enough moisture for vegetation. In this way, the overburden can be restored to its natural state by supporting vegetation. Over the years, several large scale soil cover (reconstructed watersheds) experiments are being conducted to assess the performance of different reclamation strategies by studying the basic mechanisms that control the moisture movement within these covers.

Figure 1-1 (modified after Jutla, 2006) shows the overall framework of this research program which is founded on the ongoing program of extensive monitoring. Initially, modeling of the reconstructed watersheds as a partially understood system is undertaken based on both the mechanistic and inductive modeling approaches. The knowledge gained from these approaches, supplemented by the knowledge gained from a comparison with natural systems, can be encapsulated together to formulate a decision analysis (DA) approach, entailing comprehensive and detailed sensitivity and uncertainty analyses. Based on the DA approach and the system understanding, feedback to the monitoring program and the modeling exercise can be provided. Re-directed monitoring and refined modeling will help achieve the desired comprehensive understanding of the system of reconstructed watersheds. Finally, the system understanding can be quantified towards modifying existing regulations and reclamation practices to develop sustainable reclamation strategy.

The overall framework of this research program has the following specific tasks to be completed:

1. Develop an in-house watershed simulation methodology using system dynamics modeling approach that can provide an understanding of the dynamics of the reconstructed watersheds. This task has been completed (Jutla, 2006)

2. Simulate the reconstructed watersheds using some of the readily available watershed models (e.g., HSPF, SLURP), and compare their performance with the

model developed in Task 1. The purpose of such a comparison is to gauge the utility of different modeling approaches in modeling the dynamics of reconstructed watersheds in sub-humid regions;

3. Evaluate the added gains, if any, of adopting inductive modeling approach for modeling the different components of the hydrological cycle;

4. Develop analogous models for natural watersheds, and compare their performance with that of the reconstructed watersheds. This comparison can help in possibly identifying and infilling the knowledge gap, if any, in characterizing the reconstructed watersheds with regard to their evolution over time;

5. Conduct a comprehensive study on identifying the different sources of uncertainty, and finding methods and tools to effectively incorporate uncertainty analysis into the watershed model building exercise.

6. Develop an integrated or hybrid modeling approach that benefits from the knowledge gained by adopting mechanistic and inductive modeling approaches. The objective of this task is to develop and propose to both industry and scientists the best possible tools for modeling reconstructed watersheds; and

7. Develop a multi-criterion decision analysis (MCDA) framework that can evaluate different reclamation alternatives. The objective of this task is to encapsulate the

knowledge gained with regard to reconstructed watersheds into a decision analysis tool, which can be adopted in orienting the future reclamation strategies;

The scope of this research is constrained to evaluating the utility of inductive modeling approach for modeling the hydrological processes (Task 3), and addressing the issue of uncertainty analysis in system-theoretic models for characterizing the hydrological processes (Task 5). This is identified by means of a dashed-line in Figure 1-1. The applications that are considered in this thesis are not restricted to the oil sands. Other relevant real-world applications are included for strengthening the presentation. It should be noted that this research may not address all the pertinent issues with regard to Task 3 and Task 5 in depth. Nevertheless, this research would serve as a "catalyst" for future studies in this direction, by exploring multiple avenues for accomplishing Task 3 and Task 5. Currently, two other theses are in progress to address Task 4 and Task 6.

1.6 Synopsis of the thesis

The order of chapters in this thesis is in accordance with the research objectives. Chapters Two and Three address the first objective, while Chapters Four and Five addresses the second objective. Specifically, the different parts of this thesis are presented as follows:

Chapter 2: In this chapter, a modular neural network model is proposed, and compared with a traditional neural network model. The ability of the proposed modular neural

network models in identifying the patterns in the input-output space is elucidated by applying them to streamflow modeling and actual evapotranspiration modeling.

Chapter 3: This chapter highlights the utility of adopting GP as a tool for characterizing the hydrological processes. The transparency achieved by adopting the GP paradigm, as against other system-theoretic models, is emphasised by applying GP for actual evapotranspiration modeling.

Chapter 4: Improvement in the reliability achieved by adopting a local scale model, as against a more general global scale model is elucidated in this chapter. Also, this chapter underscores the usefulness of adopting a boosting algorithm as against the conventional bagging algorithm for addressing the uncertainty in pedotransfer function (PTFs) development.

Chapter 5: This chapter emphasizes the advantages of adopting GP for PTFs development. A methodology for improving the reliability of the PTFs by accounting for the model structure uncertainty is proposed in this study.

Chapter 6: A brief summary of the thesis is given. The different levels of contribution of this thesis are highlighted. Also, the scope for further studies and research work and some implied limitations of this research are concisely discussed.

1.7 References

ASCE Task Committee on Artificial Neural Networks in Hydrology (ASCE). (2000a). "Artificial neural networks in hydrology, I, Preliminary concepts." *Journal of Hydrol. Engg.*, 5(2), 115-123.

ASCE Task Committee on Artificial Neural Networks in Hydrology (ASCE) (2000b). "Artificial Neural Networks in Hydrology, II, Hydrologic applications." *Journal of Hydrol. Engg.*, 5(2), 124-137.

Barbour, S. L., Boese, C., and Stolte, B. (2001). "Water balance for reclamation covers on oilsands mining overburden piles." In Proceedings of the 54th Canadian Geotechnical Conference, Calgary, Alta., 16-19 September 2001. Canadian Geotechnical Society, Alliston, Ont., 313-319.

Boese, K. (2003). "The design and installation of a field instrumentation program for the evaluation of soil-atmosphere water fluxes in a vegetated cover over saline/sodic shale overburden." MSc thesis, University of Saskatchewan, SK, Canada.

Duan, Q., Sorooshian, S., and Gupta, V. K. (1992). "Effective and efficient global optimization for conceptual rainfall runoff models." *Water Resour. Res*., 28(4), 1015-1031.

Eykhoff, P. (1974). *System identification: Parameter and state estimation*, John Wiley and Sons, London, UK.

Hsu, K., Gupta, V. H., and Sorooshian, S. (1995). "Artificial neural network modeling of the rainfall-runoff process." *Water Resour. Res*., 31(10), 2517-2530.

Jain, A., Sudheer, K. P., and Srinivasulu, S. (2004). "Identification of physical processes inherent in artificial neural network rainfall runoff models." *Hydrological Processes*, 118(3), 571-581.

Jutla, A. (2006). "Hydrologic modeling of reconstructed watersheds using a system dynamics approach." MSc thesis, Dept. of Civil and Geological Engineering, University of Saskatchewan, Saskatoon, SK, Canada.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection*, MIT Press, Cambridge, MA, USA.

Maier, H. R., and Dandy, G. C. (1997). "Determining inputs for neural network models of multivariate time series." *Microcomput. Civ. Eng*., 12, 353-368.

Maier, H., and Dandy, G. (2000). "Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications." *Environ. Modell. Software*, *15*(1), 101-124.

McCulloch, W. S., and Pitts, W. (1943). "A logical calculus of ideas immanent in nervous activity." *Bull. Mathematical Bio-physics*, 5, 115-133.

Penman, H. L. (1961). "Weather, plant and soil factors in hydrology." *Weather*, 16, 207-219.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). "Learning internal representations by error propagation", In Rumelhart, D. E., and McClelland, J. L., (Eds.) Parallel Distributed Processing, 1, 318-362.

Shamseldin, A. Y., Nasr, A. E., and O'Connor, K. M. (2002). "Comparison of different forms of the multi-layer feed-forward neural network method used for river flow forecasting." *Hydrol. Earth Syst. Sci*., 6, 671-684.

Singh, V.P. (1995). *Computer models of watershed hydrology*. Water Resources Publications, LLC, U.S.A.

Sudheer, K. P., and Jain, A. (2004). "Explaining the internal behaviour of artificial neural network river flow models." *Hydrological Processes*, 118(4), 833-844.

Werbos, P. (1974). "Beyond regression: New tools for prediction and analysis in the behavioral sciences." PhD thesis, Dept. of Applied Mathematics, Harvard University, Cambridge, MA, USA.

Wilby, R. L., Abrahart, R. J., and Dawson, C. W. (2003). "Detection of conceptual model rainfall-runoff processes inside an artificial neural network." *Hydrological Sciences Journal*, 48(2), 163-181.

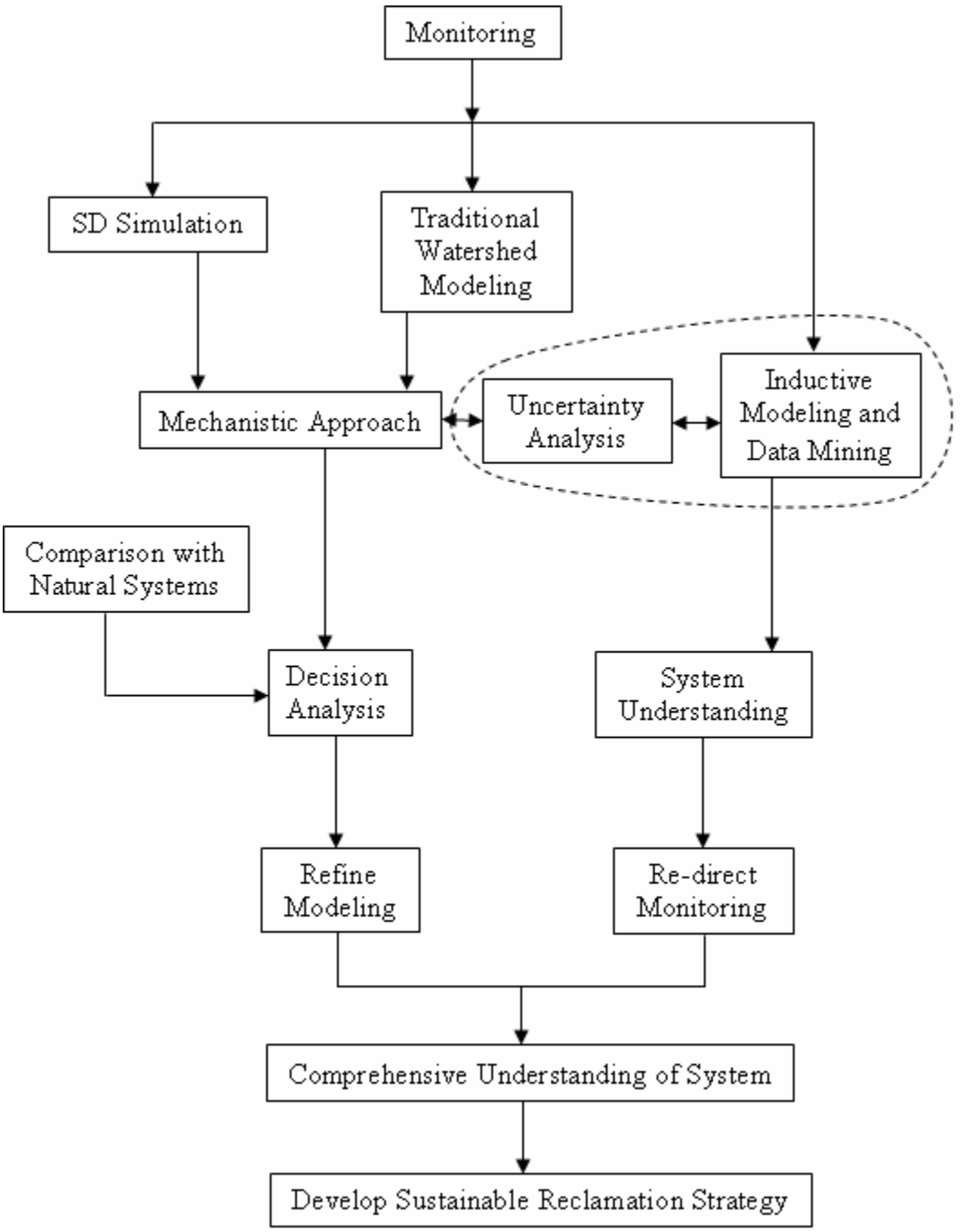

Figure 1-1 Framework of the research program for developing a sustainable reclamation strategy

Chapter 2 - Spiking-Modular Neural Networks: A Neural Network Modeling Approach for Hydrological Processes

A similar version of this chapter has been copyrighted and published in the Water Resources Research.

Citation: Parasuraman, K., Elshorbagy, A., and Carey, S. K. (2006). "Spiking modular neural networks: A neural network modeling approach for hydrological processes." *Water Resour. Res.,* 42, W05412, doi: 10.1029/2005WR004317.

Contribution of the PhD candidate

Model conceptualization and computer program development were carried out by Kamban Parasuraman, with Dr. Amin Elshorbagy and Dr. Sean Carey providing advice on various aspects of the work. The text of the published paper was created by Kamban Parasuraman with Dr. Amin Elshorbagy and Dr. Sean Carey critically reviewing the manuscript.

Contribution of this chapter to the overall study

This work was aimed at extending the utility of the neural networks beyond forecast applications as a tool for scientific investigations. The hypothesis was that associating the self-organizing networks with the modular networks would help in bringing transparency to the way by which neural networks identify the patterns in the input-output space. A modular neural network model called the spiking modular neural networks (SMNNs) is proposed in this study, which first identifies the patterns in the

input space, before developing individual models to associate the identified patterns in the input space with their corresponding patterns in the output space. The performance of the proposed models is evaluated using two-distinct case-studies, namely, (i) streamflow modeling, and (ii) evapotranspiration modeling. The SMNNs are shown to be effective in discretizing the complex mapping space into simpler domains that can be learned with relative ease.

2.1 Abstract

Artificial Neural Networks (ANNs) have been widely used for modeling hydrological processes that are embedded with high nonlinearity in both spatial and temporal scales. The input-output functional relationship does not remain the same over the entire modeling domain, varying at different spatial and temporal scales. In this study, a novel neural network model called the spiking-modular neural networks (SMNNs) is proposed. An SMNN consists of an input layer, a spiking layer, and an associator neural network layer. The modular nature of the SMNN helps in finding domain dependent relationships. The performance of the model is evaluated using two distinct case studies. The first case study is that of streamflow modeling and the second case-study involves modeling of eddy-covariance (EC) measured evapotranspiration. Two variants of SMNNs were analyzed in this study. The first variant employs a competitive layer as the spiking layer and the second variant employs a self-organizing map (SOM) as the spiking layer. The performance of SMNNs is compared to that of a regular feed-forward neural network (FF-NN) model. Results from the study demonstrate that SMNNs performed better than FF-NNs for both the case studies. Results from partitioning analysis reveal that, compared to FF-NNs, SMNNs are effective in capturing the dynamics of high flows. In modeling evapotranspiration, it is found that net-radiation and ground temperature alone can be used to model the evaporation flux effectively. The SMNNs are shown to be effective in discretizing the complex mapping space into simpler domains that can be learnt with relative ease.

2.2 Introduction

2.2.1 Advances in Neural Network Modeling in Hydrology

Modeling of hydrological processes is central for efficient planning and management of water resources, which is usually achieved either by conceptual models or by systems theoretic models. Artificial neural networks (ANNs), a systems theoretic method, have been shown to be a promising tool for modeling hydrological processes (ASCE Task Committee on the Application of Neural Networks in Hydrology, 2000a; Maier and Dandy, 2000). The increasing utility of ANNs in modeling hydrological processes is attributed to their ability to capture complex nonlinear relationships between inputs and outputs with an incomplete understanding of the physics of the process involved.

The presence of discontinuity in rainfall-runoff mapping of a watershed and significant variations in input space motivated Zhang and Govindaraju (2000) to develop modular neural networks. They attributed the discontinuity to rainfall-runoff functional relationships being different for low, medium, and high magnitudes of streamflow. The modular neural network developed by Zhang and Govindaraju (2000) consists of a gating network and a series of neural networks. Each neural network in this series is termed as an expert, mapping the relationship in a subset of input space. The gating network helps in identifying the expert for a given input vector. The gating network outputs the probability of an input vector association with each of the experts. The output from the network is then calculated by multiplying the individual expert's response by the corresponding weights (probability) of the gating network. Zhang and Govindaraju

(2000) showed that the performance of modular neural networks is better than that of the regular feed-forward neural networks (FF-NNs). Hsu et al. (2002) developed a self-organizing linear output map (SOLO); an artificial neural network model studying the rainfall-runoff modeling problem. SOLO consists of a classification layer and a mapping layer. Classification of input space is achieved by means of a self-organizing feature map (SOFM) (Kohonen, 1989). The mapping layer helps map the input to its corresponding output by means of piecewise linear regression functions. Based on their study, Hsu et al. (2002) concluded that the SOLO model resulted in rapid and precise estimation of system outputs. Hong et al. (2005) extended SOLO to a SONO (self-organizing nonlinear output) model for cloud patch-based rainfall estimation. Similar to SOLO, SONO made use of SOFM in its classification layer. However, in SONO, mapping input space to output is achieved by means of nonlinear regression. Recently, Bowden et al. (2005) used SOM to reduce the dimensionality of the input space and obtain independent outputs, with the objective of finding the optimal combination of input parameters for neural networks modeling. The input variables are presented to the SOM and only one input is selected from each cell based on its proximity to the cluster centers. These selected inputs are then used to train the neural networks models. Based on the above studies, it can be concluded that input-output functional relationship is quite different in different domains of the input space. Hence, cluster-based mapping appears to be a promising alternative to FF-NNs, particularly in cases of processes where the input-output functional relationship is fragmented or discontinuous. Although Zhang and Govindaraju (2000) demonstrated the importance of modular neural networks and Hsu et al. (2002) demonstrated the utility of self-organizing maps in modeling hydrological processes, little effort has been made to

study the usefulness of harnessing both modular learning and self-organizing networks. Moreover, to the knowledge of the authors, no work has been reported in literature to compare the different ways by which self-organization in networks can be achieved. This comparison is of particular interest as it helps in identifying the proper self-organization technique suitable for modular learning.

2.2.2 Neuro-Hydrology: Beyond Rainfall-Runoff Modeling

Compared to other hydrological processes such as rainfall and runoff, evaporation (used here to describe latent heat flux from the surface) is more dynamic because it involves continuous exchange of water molecules between the land and the atmosphere. Hence, the evaporation process is embedded with huge variability in both spatial and temporal scales. For this reason, evaporation is the least satisfactorily explained component of the global hydrological cycle (Sudheer et al., 2002). An improvement in the estimates of evaporation helps in partitioning the available moisture into (1) water loss back to the atmosphere, and (2) the water available for runoff. Although water-balance components including rainfall, infiltration, and runoff are measured directly, evaporation is most commonly estimated by energy balance, mass transfer, or water budget methods (Sudheer et al., 2002). Traditionally, pan evaporation is used as an index for free water surface (lakes and reservoirs) evaporation, and empirical coefficients are applied to correlate pan evaporation to reference crop evapotranspiration (ET_o). Alternatively, lysimeters are used to directly estimate surface or crop evapotranspiration (ET_c) by measuring changes in mass of a control volume (Singh, 1989). Measurements of evaporation by pan-evaporimeter and lysimeter are subject to a

large set of assumptions, cumbersome and labor-intensive, and may not be appropriate for large-scale studies. In research applications, micrometeorological methods such as energy-balance-Bowen-ration (EBBR) and eddy-covariance (EC) are typically used to measure actual evaporation (ET) (Drexler et al., 2004). However, these methods are expensive and are sufficiently complex to limit their widespread application. In order to overcome these problems, numerous studies have been carried out to estimate ET_o from climatic data. Key examples of such studies include (i) empirical relationships between meteorological variables (Holdridge, 1962; Stephens and Stewart, 1963; Blaney-Criddle, 1950; Linacre, 1977; Thornthwaite, 1948; Priestley and Taylor, 1972; Hargreaves and Samani, 1982) and (ii) physically-based equations (Penman, 1948; Monteith, 1965). While the former methods estimate ET_o based on climatological data, the latter methods link evaporation dynamics with the supply of net-radiation and aerodynamics transport characteristics of a natural surface, and hence are termed combination methods.

The success of the neural network models in modeling different hydrological processes provides an impetus to test the applicability of neural networks in modeling the highly dynamic evaporation process. Key examples of such studies include Sudheer et al., (2002); Kumar et al., (2002); Sudheer et al., (2003); Trajkovic et al., (2003). Most of the above studies on modeling evaporation using neural networks estimated either of PM estimates of evaporation (Kumar et al., 2002; Trajkovic et al., 2003), of the pan (Kumar et al., 2002 and Sudheer et al., 2002), or of the lysimeter (Sudheer et al., 2003) measured values. To the knowledge of the authors, no work has been reported in the literature to address the application of neural networks in modeling EC measured evaporation flux.

In this study, a novel neural network model is proposed: the spiking-modular neural network (SMNN). The SMNN is based on the concepts of both self-organizing networks and modular networks. The performance of the model is tested on two diverse case studies. The first case study involves modeling of streamflows and the second case study involves modeling of actual evaporation measured via eddy-covariance (EC). While the first case study represents a single-input-single-output (SISO) process, the second case study represents a multiple-input-single-output (MISO) process. Two variants of SMNNs; one employing competitive learning in the spiking layer and the other employing self-organizing maps in the spiking layer are tested. The specific objectives of this research are as follows: (1) to evaluate the performance of regular FF-NN in modeling streamflows and EC measured evaporation flux; (2) to compare the performance of FF-NNs with the proposed SMNNs on both case studies; and (3) to provide insight into the performance of the SMNNs.

The remaining part of this chapter is organized as follows. In section 2.3, an introduction to neural networks is given. Section 2.4 presents the architecture of the spiking-modular neural networks adopted in this study. Streamflow estimation and modeling of evaporation flux are discussed in sections 2.5 and 2.6, respectively. Results and discussion are presented in section 2.7, and the final section summarizes important research conclusions.

2.3 Neural Networks

ANNs are a method of computation and information processing motivated by the functional units of the human brain, namely neurons. According to Haykin (1999), a neural networks is a massively parallel distributed information processing system that is capable of storing the experiential knowledge gained by the process of learning, and of making it available for future use. Mathematically, ANNs are universal approximators with an ability to solve large-scale complex problems such as time series forecasting, pattern recognition, nonlinear modeling, classification, and control. This is achieved by identifying the relationships among given patterns.

FF-NNs are the most widely adopted network architecture for the prediction and forecasting of water resources variables (Maier and Dandy, 2000). Typically, FF-NNs consist of three layers: input layer, hidden layer, and output layer. The number of nodes in the input layer corresponds to the number of inputs considered for modeling the output. The input layer is connected to the hidden layer with weights that determine the strength of the connections. The number of nodes in the hidden layer indicates the complexity of the problem being modeled. The hidden layer nodes consist of the activation function, which helps in nonlinearly transforming the inputs into an alternative space where the training samples are linearly separable (Brown and Harris, 1994). The hidden layer is connected to the output layer. An epoch is the presentation of the whole training samples to the neural networks model. Detailed review of ANNs and their application in hydrology can be found in Maier and Dandy (2000) and in ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000 a, b).

The structure of the three-layered FF-NN used in this study is shown in Figure 2-1. The neural network model consists of *'j'* input neurons, *'k'* hidden neurons, and *'l'* output neurons. Symbolically, the ANN architecture shown in Figure 2-1 can be represented as *ANN(j,k,l)*. The FF-NN adopted in this study makes use of the tan-sigmoidal activation function in the hidden layer and the linear activation function in the output layer. In Figure 2-1, W_{kj} represents the connection weight between the j^{th} input neuron and k^{th} hidden neuron. Similarly, W_{lk} represents the connection weight between the k^{th} hidden neuron and l^{th} output neuron. Parameters b_k and b_l represent the bias of the corresponding hidden and output layer neurons. If x_j represents the input variables and y_l represents the output variable, then the inputs are transformed to output by the following equations:

$$y_l = f_1\left[\sum_{k=1}^{K} w_{lk} f_2\left(\sum_{j=1}^{J} w_{kj} x_j + b_k\right) + b_l\right] \tag{2.1}$$

$$f_2(p) = \frac{2}{\left(1 + e^{-2p}\right)} - 1 \tag{2.2}$$

where $f_1(.)$ represents the linear activation function and $f_2(.)$ represents the tan-sigmoidal activation function. While the tan-sigmoidal activation function squashes the input between -1 and 1, the linear activation function calculates the neurons output by simply returning the value passed to it. One of the important issues in the development of neural networks model is the determination of an optimal number of hidden neurons that can satisfactorily capture the nonlinear relationship existing between the input variables and the output. The number of neurons in the hidden layer is usually determined by trial-

and-error method with the objective of minimizing the cost function (ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000a).

The typical cost function used in training FF-NNs involves minimizing the mean sum of squares of the network errors (MSE). However in this study, in order to overcome the problem of overfitting, a Bayesian-regularization back propagation algorithm (Demuth and Beale, 2001) is used for training the FF-NNs. The Bayesian-regularization back propagation algorithm improves the generalization property of the ANN model by developing networks with smaller weights and biases, and thus a smoother response that is less likely to result in overfitting (Demuth and Beale, 2001). Hence along with minimizing MSE, the cost function in Bayesian-regularization back propagation algorithm (Equation 2.3) involves minimizing the mean of the sum of squares of the network weights and biases (MSW). In Equation 2.3, y_i and $y_i^{'}$ represent the measured and computed counterparts; n, and N represents the number of training instances and the number of network parameters respectively. The success of the regularization depends on the choice of an appropriate value of the regularization parameter, α. In this study, the method by MacKay (1992) is adopted, where the optimal α is determined in a Bayesian framework using automatic relevance determination.

$$MSE\_REG = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^{'}\right)^2 + (1-\alpha)\left(\frac{1}{N}\sum_{j=1}^{N} w_j^2\right) \tag{2.3}$$

A systematic search of different network configuration and user-adjustable parameters was carried out to ascertain the optimal network architecture, with the objective of minimizing the cost function. The optimal network architecture is the one which results in the least cost function. Although the FF-NNs are capable of approximating a continuous well-behaved relationship between the input and output variables, they may not be suitable for mapping a fragmented or discontinuous representation of the training data that has significant variation over the input space (Zhang and Govindaraju, 2000). In SMNNs, explained below, the above problem is overcome by decomposing the complex mapping space into simpler sub-domains that can be learned with relative ease by the individual FF-NN models.

2.4 Spiking Modular Neural Networks (SMNNs)

The structure of the SMNNs is shown in Figure 2-2. The input layer consists of j input neurons $(x_1, x_2, x_3, \ldots, x_j)$ where number of input neurons, j, is equal to the number of input variables. The input layer neurons are connected to the spiking layer, which serves as the *memory* of the system, learning and storing different input patterns that can be used in classifying future input vectors based on patterns learned during the training process. In the spiking layer, clustering of input space is achieved by unsupervised (or self organized) learning, which is defined as the learning process that does not involve a teacher or critic to oversee the learning. Self-organized learning consists of repeatedly modifying the synaptic weights of a neural network in response to activation patterns and in accordance with prescribed rules, until a final configuration appears (Haykin, 1999).

Furthermore, self-organizing networks can learn to detect regularities and correlations in the input space, and accordingly adapt their future responses to that input.

Self-organization in networks can be achieved in two ways: (1) competitive learning; and (2) self-organizing maps (SOMs) (Demuth and Beale, 2001). In competitive learning, the neurons of the network compete among themselves to be active (spike), the winner of which is called a *winner-takes-all* neuron (Haykin, 1999). The SOMs are a special case of self-organizing system as they learn to recognize groups of similar input vectors in such a way that neurons physically near each other in the neuron layer respond to similar input vectors. A SOM is therefore characterized by the formation of a topographic map of the input patterns in which the spatial locations of the neurons in the lattice are indicative of intrinsic statistical features contained in the input patterns (Haykin, 1999). Hence, the main difference between competitive learning and SOMs is that, while the former learns only the distribution, the latter learns both the distribution and the topology (neighboring neurons) of the input space. SOMs are either 1-dimensional or 2-dimensional and the structure of SOMs is usually represented by the form $n_1 \times n_2$, where n_1 and n_2 represent the number of rows and column of neurons respectively. The following paragraph outlines the mechanism involved in learning patterns by the self-organizing networks.

The weights of the self-organizing networks are initialized to the center of the input ranges. Once initialized, the self-organizing network neurons are trained by the Kohonen learning rule (Kohonen, 1989) to identify the clusters in the input space, and

allow the connection weights of the neuron to learn an input vector. Each neuron of the self-organizing network competes to respond to an input vector. Proximity of inputs to each neuron is determined based on Euclidean distance (d_c) as given in Equation (2.4):

$$d_{c=1..m} = \left[\sum_{j=1}^{J} \left(x_j - w_{cj} \right)^2 \right]^{0.5} \tag{2.4}$$

where m denotes the number of clusters and w_{cj} represents the connection weight linking j^{th} input variable and c^{th} neuron of self-organizing networks. In the case of competitive learning, the neuron whose weight vector is closest to that of the input vector is updated to be even closer. However, in self-organizing maps, along with the closest neuron, the neurons in the neighborhood on the closest neuron are also updated to be even closer. The result of such training results in a neural network model where the winning neuron is more likely to win the competition the next time a similar vector is presented, and less likely to win when a very different input vector is presented. Hence for a given input vector, the neuron which represents the cluster that is closest to the input vector outputs 1 (spikes), while the remaining neurons output 0. More information on self-organizing networks can be found in Demuth and Beale (2001) and Kohonen (1989).

Once classification of the input space is achieved, mapping of inputs to the corresponding outputs has to be carried out. Mapping inputs to outputs can be achieved by either linear-regression or FF-NNs. In the case of highly correlated input variables, use of the linear regression model for mapping inputs to outputs may require conversion

of input variables to its principal components in order to avoid colinearity problem (Hsu et al., 2002). In this study, mapping of inputs to outputs is achieved by neural networks and as these networks associate input patterns to outputs. They are termed *associator* neural networks. The associator neural networks are similar to the neural networks detailed in section 2.2. SMNNs belong to a class of modular neural networks as the SMNNs works by developing *'c'* different associator neural networks, each specializing in *'c'* different subsets of the mapping domain.

In this study, the performance of the proposed SMNNs employing the self-organizing networks (both competitive and SOMs) is tested. The first SMNNs makes use of the competitive network as the spiking layer and herein it will be referred as SMNN(Competitive). The second variant of the SMNNs makes use of SOM as the spiking layer. Herein, the SMNNs with SOM as spiking layer will be referred as SMNN(SOM). Since competitive networks learn only the distribution of the input space and SOMs learn both the distribution and topology of the input space, comparison of SMNN(Competitive) and SMNN(SOM) would help understand the effect of topology learning in SMNNs performance. Since there is no theoretical principle to determine the optimum size of the Kohonen layer (Cai et al., 1994), the number of nodes in the spiking layer of SMNN(Competitive) and SMNN(SOM) is determined by trial-and-error method. Starting with two nodes in each of the spiking layer and hidden layer, the optimal architecture of the SMNNs is evaluated by performing a systematic search over different network configurations with the objective of minimizing the cost function (Equation 2.3). Symbolically, the optimal architecture of a SMNN with *c* spiking neurons can be

represented as SMNN[*c*, *ANN(j,k,l)]*, where *ANN(j,k,l)* represents the optimal associator neural networks configuration.

2.5 Streamflow Prediction

The monthly streamflow values of the English River, Ontario, Canada, between Umfreville (49° 52′ N, 91° 27′ W) and Sioux Lookout (50° 4′ N, 91° 56′ W) is considered in this study. Umfreville is located upstream from Sioux Lookout. The streamflow values are obtained from Environment Canada's Hydrometric Database (HYDAT) (Government of Canada, available online http://www.msc.ec.gc.ca/wsc/hydat/H2O/index_e.cfm, 2004). Flow values at Sioux Lookout are considered missing and are estimated based on the flow values at Umfreville. Out of the available data between January 1924 and December 1981, approximately 70% of the data is used for training and the remaining 30% of the data is used for testing the developed model. (i.e., monthly flow values from January 1924 to August 1965 are used for training the neural networks, and the flow values from September 1965 to December 1981 are considered for testing the models.) The statistical properties of the entire dataset along with the statistical properties of the datasets used for training and testing are presented in Table 2-1. The flow at Sioux Lookout shows slightly less variability than the flow at Umfreville. Both training and testing datasets have similar statistical properties.

2.5.1 Predicting Streamflow Using FF-NNs

A three-layered FF-NN is considered in this study. The input layer consists of a single input neuron representing the flow at Umfreville. The output layer has a single

output neuron corresponding to the flow at Sioux Lookout. Mapping of inputs to outputs is achieved by the hidden layer neurons. The number of hidden layer neurons is determined by the trial-and-error method as detailed in section 2.2, and the optimal number of hidden neurons is three. Hence, the neural network architecture adopted in this study is of the form ANN(1,3,1). For this study, 2000 epochs is found optimal for training the networks.

2.5.2 Predicting Streamflow Using SMNNs

Similar to the FF-NN, the SMNN model consists of single input and single output neurons. The performances of both SMNN(Competitive) and SMNN(SOM) in streamflow prediction are evaluated in this study. Similar to the method for determining the number of hidden nodes in FF-NNs, the numbers of neurons in the spiking layers of both the SMNN(Competitive) and SMNN(SOM) models are determined by the trial-and-error method as detailed in section 2.3. The optimal number of neurons for both variants of SMNNs is two, indicating that there are two different clusters in the input space. As a next step, the clustered input space is mapped to the corresponding output space. This is achieved by the associator neural networks. Since there are two different clusters, the SMNNs consist of two associator neural networks. Each of these associator neural networks specializes in mapping the input-output relationship at the respective domain of the mapping space.

The spiking layer is trained until the neurons in this layer are able to learn the classification of the input space. This is determined by finding the number of epochs beyond which there is no further improvement in classification of input vectors. For this

study, the optimal number of epochs required for training the spiking layer is found to be 300. The architecture of all the associator neural networks is similar and is determined in a way analogous to the method used to determine the architecture of the regular FF-NNs. For both variants of SMNNs, ANN(1,3,1) is the optimal architecture of the associator neural networks. Symbolically, the optimal architecture of SMNNs in modeling streamflow is given by SMNN[*2, ANN(1,3,1)*].

2.6 Modeling Actual Evaporation Using Climatic Data

2.6.1 Site Description and Data

South Bison Hill (SBH) (57° 39' N and 111° 13' W), a overburden pile located north of Fort McMurray, Alberta, Canada, is considered in this study. SBH was constructed with waste-rock material from oilsands mining in stages between 1980 and 1996. The area of SBH is 2 km^2, rises 60 m above the surrounding landscape and has a large flat top several hundred meters in diameter. To reclaim the overburden so that revegetation can occur, the underlying shale is covered by a 0.2 m layer of peat on top of a 0.8 m layer of till. The top of SBH is dominated by foxtail barley (*Hordeum jubatum*); also present are other minor species such as fireweed (*Epilobium angustifolium*). Estimation of evaporation from the reconstructed watershed is of vital importance as it plays a major role in water-balance of the system, which links directly to ecosystem restoration strategies.

Micrometeorological techniques were used to directly measure evaporation and the surface energy balance. A mast located in the approximate center of SBH was equipped to measure air temperature (AT) and relative humidity (RH) (HMPFC, Vaisala, 3 m) housed in a Gill radiation shield, ground temperature (GT) (TVAC, Campbell Scientific, averaged 0.01-0.05 m depth), all-wave net radiation (R_n) (CNR-1, Kipp and Zonen, 3 m), and wind speed (WS) (015A Met One, 3.18 m). All instruments were connected to a datalogger (CR23X, Campbell Scientific) sampled at 10 seconds and an average or a cumulate record was logged every half-hour. The energy balance of the surface is given by:

$$R_n = LE + H + G + \varepsilon \tag{2.5}$$

Where *LE* is the latent heat flux (evaporation when divided by the latent heat of vaporization), *H* the sensible heat flux, *G* the ground heat flux and ε the residual flux density, all expressed in W m^{-2}. *G* was measured using a CM3 ground heat flux plate (REBS) placed at 0.05 m depth. LE and H were measured directly via the open-path eddy covariance (EC) technique (Leuning and Judd, 1996) using a CSAT3 sonic anemometer (Campbell Scientific) and an LI-7500 CO_2/H_2O gas analyzer (Li-Cor) with the midpoint of the sonic head located on a boom 2.8 m above the ground surface. Measurements of *H* and *LE* were taken at 10 Hz and fluxes were calculated using 30 minute block averages with 2-D coordinate rotation. Sensible heat fluxes were calculated using the sonic virtual temperature (Schotanus et al., 1983) and latent heat fluxes were corrected for changes in air density (Webb et al., 1980). Fluxes were removed when friction velocity was less

than 0.1 m/s due to poor energy balance closure at low wind speeds (Twine et al., 2000; Baker and Griffis, 2005). Flux measurements were also removed during periods of rainfall and during periods of unexpected change in state variables. No gap filling was performed.

Variation of evaporation is commonly perceived as highly dependent on climatic variables such as temperature, humidity, solar radiation, and wind speed (Brutsart, 1982; Sudheer et al., 2003). Hence in this study, the climatic variables AT, GT, R_n, RH, and WS, which are commonly measured at weather stations, are used to estimate the evaporation flux measured by the EC system. As a common practice, a training set is used for model development and an independent validation set is used to test the efficiency of the developed model. Hourly data between May 20, 2003, and June 9, 2003, comprise the training set and the data between June 18, 2003, and June 28, 2003, comprise the testing set. The training set consists of 500 instances while the testing set consists of 247 instances. Plots showing the correlation of input variables AT, GT, R_n, and RH with LE are presented in Figure 2-3. The correlation plot between WS and LE is not shown as there is no significant correlation between them. The correlation plots shown in Figure 2-3 are based on the training set alone. As expected, air temperature (R^2 = 0.227), ground temperature (R^2 = 0.405), and net radiation (R^2 = 0.569) are shown to have a positive trend with LE, while relative humidity (R^2 = 0.114) has a negative relationship with LE.

Traditionally, Penman-Monteith is the most widely used method for estimating evapotranspiration due to the widespread availability of the input variables. The hourly FAO Penman-Monteith (Temesgen et al., 2005) equation is given by Equation (2.6):

$$ET_0 = \frac{0.408\Delta(R_n - G) + \gamma \frac{37}{AT + 273} WS(e^0 - e^a)}{\Delta + \gamma(1 + 0.34WS)} \tag{2.6}$$

where, R_n is net radiation at the grass surface (MJ m^{-2} hr^{-1}), G is soil heat flux density (MJ m^{-2} hr^{-1}), Δ is the saturation slope vapour pressure curve at AT (K Pa $^oC^{-1}$), γ is the psychrometric constant (K Pa $^oC^{-1}$), e^0 is saturation vapour pressure at air temperature AT (K Pa), e^a is the average hourly actual vapour pressure (K Pa), and WS is the average hourly wind speed (m/s). It should be noted that the evaporation calculated by the Penman-Monteith equation is potential evaporation for a well-watered surface, and not actual evaporation. Several methods of converting ET_0 to actual evaporation have been illustrated by Saxton (1981) and Jensen (1981), which estimate actual evaporation based on water balance or by empirical equations. Eddy covariance (EC) offers a convenient way to directly measure actual evaporation and hence, in this study, an attempt has been made to model EC-measured evaporation flux using neural networks.

2.6.2 Estimation of Evaporation Flux Using FF-NNs

The FF-NN model considered for modeling evaporation flux consists of five input neurons, representing AT, GT, R_n, RH, and WS. The output layer consists of a single neuron representing LE. As explained in section 2.2, the optimal number of hidden

nodes is found by the trial-and error method, and is found to be four. Hence, the neural network architecture adopted in this study is of the form ANN(5,4,1). Bayesian-regularization algorithm is used for training the networks. For this case-study, 5000 epochs is found optimal for training the FF-NNs.

2.6.3 Estimation of Evaporation Flux Using SMNNs

The performances of both variants of SMNNs (SMNN(Competitve) and SMNN(SOM)) are tested with regard to estimating the EC-measured evaporation flux. The SMNNs considered in this application consist of five input neurons. By trial-and-error method, as detailed in section 2.3, the optimal number of neurons in the spiking layer was found to be eight for both SMNN(Competitive) and SMNN(SOM). The spiking layer consists of eight neurons, representing individual clusters in the input space. Eight hundred epochs are found optimal for training the spiking layer. Corresponding to each cluster, eight different associator neural network models specializing in mapping input-output relationships at different domains of the mapping space are constructed. The associator neural network models employ Bayesian-regularization algorithm for training the networks. The optimal network architecture of associator neural networks is ANN(5,4,1). Symbolically, the optimal architecture of SMNNs can be represented as SMNN[*8, ANN(5,4,1)*].

2.7 Results and Discussions

Since RMSE and MRE give different details about the predictive ability of the models (Karunanithi et al., 1994), a multi-criterion performance evaluation is carried out. For both case studies, the performances of the different models are evaluated based on: (i) root mean square error (RMSE), (ii) mean absolute relative error (MRE), and (iii) coefficient of correlation (R). RMSE, MRE, and R are calculated using Equations (2.7), (2.8), and (2.9) respectively, where n represents the number of instances presented to the model; y_i and $y_i^{'}$ represent measured and computed evaporation flux respectively; and $\overline{y}$ represents the mean of the corresponding variable:

$$RMSE = \left[\frac{1}{n} \sum_{i=1}^{n} (y_i - y_i^{'})^2 \right]^{0.5} \tag{2.7}$$

$$MRE = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - y_i^{'}}{y_i} \right) \tag{2.8}$$

$$R = \frac{\sum_{i=1}^{n} \left(y_i - \overline{y_i} \right) \left(y_i^{'} - \overline{y_i^{'}} \right)}{\sqrt{\sum_{i=1}^{n} \left(y_i - \overline{y_i} \right)^2 \sum_{i=1}^{n} \left(y_i^{'} - \overline{y_i^{'}} \right)^2}} \tag{2.9}$$

2.7.1 Streamflow Modeling

Table 2-2 presents the statistical performances of different neural network models in estimating streamflow at Sioux Lookout based on the flow at Umfreville. FF-NNs resulted in an RMSE of 30.4 m^3/s, an MRE of 0.22, and a correlation coefficient of 0.96. Both SMNN(Competitive) and SMNN(SOM) models performed better than the FF-

NN model in terms of RMSE during testing, however all the models performed on par in terms of MRE and R. Figure 2-4 indicates that different neural network models were able to mimic the trend of measured flows at Sioux Lookout. However, as will be shown later, SMNNs resulted in better prediction accuracy in the case of high-flows. The SMNN(Competitive) model performed marginally better (lower RMSE) than the SMNN(SOM).

To improve the insight into the performance of SMNNs, a plot showing the instances at which different spiking layer neurons spiked is shown in Figure 2-5. Both SMNN(Competitive) and SMNN(SOM) were successful in delineating high-flows from low-flows. The threshold value separating high-flows and low-flows differed between SMNN(Competitive) and SMNN(SOM). The solid lines in Figure 2-5 indicate the threshold values. For SMNN(Competitive), the threshold value was 185.6 m^3/sec, and for SMNN(SOM), the threshold value was approximately 145.7 m^3/sec. In SMNN(Competitive), 48 instances were above the threshold value and the remaining 148 instances were below the threshold value. The MRE of the instances above the threshold is 0.14 and the MRE of the instances below the threshold is 0.25. However, in case of SMNN(SOM), 78 instances were above the threshold value and the remaining 118 instances were below the threshold value. The MRE of the instances above the threshold is 0.19 and the MRE of the instances below threshold is 0.25.

Sajikumar and Thandaveswara (1999) and Tokar and Markus (2000) experienced problems in learning patterns using a back-propagation algorithm when

target flow values were in the neighborhood of zero. However, Minns and Hall (1996) reported that the regular FF-NNs were not able to mimic larger peaks in flow data. This problem with regard to the performance of SMNNs is analyzed by partitioning analysis. Partitioning is carried out by arbitrarily choosing a certain threshold of flow, then finding the errors (RMSE and MRE) between the measured and estimated flows, both above and below the threshold value. The testing dataset, which consists of 196 instances, is considered for this analysis. The mean (μ) and the standard deviation (σ) of the data are 121.69 m^3/s and 86.31 m^3/s respectively. A value slightly less than $\mu+\sigma$ is considered as the threshold (200 m^3/s). Out of the total 196 instances, 167 instances are below the threshold value and the remaining 29 instances are above the threshold value. The relative performance of these models above and below the threshold is presented in Table 2-3. From the table it can be seen that the performance of SMNNs is on par with FF-NNs performance below the threshold value (<200 m^3/s). However, SMNNs performed better than FF-NNs for flows above the threshold value (>200 m^3/s). Based on the above analysis, it is concluded that the proposed SMNNs are a promising alternative for modeling high flows.

2.7.2 Modeling Evaporation

Jackson et al. (1976) and Salvucci (1997) showed that evaporation is a two-stage process, with a climate-control stage followed by a soil-control stage. Different dynamics of state variables govern each of these stages. Apart from these two distinct stages, it should be noted that different combinations of evaporation, ground and sensible heat flux that with net-radiation could satisfy the energy balance (Equation 2.5). Hence the

performance of SMNNs in capturing such discontinuous input-output relationship would be of particular interest. Performance statistics of different neural network models for estimating actual ET during both training and testing are presented in Table 2-4. Modeling of evaporation flux using FF-NNs resulted in an RMSE of 73.4 W/m^2, an MRE of 1.5, and an R of 0.69. Comparing the performance of FF-NN with SMNNs, both SMNN(Competitive) and SMNN(SOM) outperformed FF-NN in estimating evaporation flux. This reiterates the fact that the input-output relationship is discontinuous over the mapping horizon and hence modular neural networks could offer a promising alternative to capture such discontinuous input-output mapping.

It can be noted, in general, that the training RMSE of the neural network models showed significant variations with the testing RMSE (Table 2-4). Considerable variability in RMSE statistics between training and testing can be attributed to the following reason. The median of LE dataset during training and testing are 34.1 W/m^2 and 61.2 W/m^2 respectively. Since median gives a measure of central tendency, it implies that, compared to the training dataset, the testing dataset is dominated more by higher values of evapotranspiration. The values of evapotranspiration in the testing dataset are roughly twice (61.2/34.1) the values of evapotranspiration in training dataset. Karunanithi et al. (1994) demonstrated that MSE and MRE provide different types of information about the predictive capability of the model. In their work, it has been shown that MSE is more sensitive to errors at high and low values, whereas MRE provides a more balanced perspective of the goodness of fit at moderate values. Since squared-error statistics give more weighting to high values, the RMSE during testing is approximately twice that

during training, preserving the ratio between the medians. This illustrates that the neural network models are not over-trained. SMNN(Competitive) resulted in an RMSE of 70.2 W/m^2, an MRE of 1.2, and an R of 0.71. However, SMNN(SOM) resulted in an RMSE of 73.0 W/m^2, an MRE of 1.3, and an R of 0.67. From Table 2-4, it can be concluded that SMNN(Competitive) provides a more generalized representation of the evaporation process. Analysis of results obtained from SMNNs reveals that different combinations of inputs may lead to the same value of evaporation flux. This was evident when similar values of evaporation flux were obtained even when different spiking layer neurons spiked (i.e., inputs from different clusters). The above effect is of particular interest as it illustrates that the SMNN as a data driven model is able to confirm that different combination of state variables can satisfy the energy-balance equation. More explanation in this regard is presented in the subsequent section of this chapter.

Figure 2-6 compares PM and SMNN(Competitive) estimates of evaporation with the EC measured evaporation flux between May 20, 2003, and June 9, 2003. Since PM estimates potential evaporation, and in reality water is not always freely available (supply limited) to evaporate, PM overestimates evaporation during supply limited conditions. Although it is not prudent to directly compare the PM estimates with the neural networks models predicting EC measured LE flux, the above comparison is made due to the following reason. Abbott et al. (1986) stated that the PM method, which accounts for the influence of vegetation on evapotranspiration, has been used frequently to model the evapotranspiration flux. The wide spread utility of PM method in characterizing evaporation is due to its ability in predicting evaporation based on readily

available climatic data like solar radiation, relative humidity, wind speed and air temperature. The potential estimates of evaporation given by the PM method can be converted to actual evaporation by considering the soil moisture limitations. However, compared to climatic data, the soil moisture data is not always readily available at the temporal resolution of other climatic variables. Furthermore, soil moisture shows large variability in both spatial and temporal scales (Entekhabi et al., 1996). Hence interpolation of the soil moisture data to match the temporal resolution of climatic variables is not prudent. Due to the above limitations, conversion of potential evaporation to actual evaporation is cumbersome and involves large uncertainty. In this regard, the utility of neural networks model in directly predicting actual evaporation from climatic data alone is tested and compared with the PM estimates that would be used otherwise. The RMSE between the measured and the PM-estimated evaporation flux is 88.2 W/m^2, which is comparatively higher than the RMSE of 32.2 W/m^2 obtained by SMNN(Competitive). For the period between June 18, 2003, and June 28, 2003 (testing data), the RMSE between measured and PM-estimated evaporation flux is 92.5 W/m^2, which is again significantly greater than the RMSE of 70.2 W/m^2 obtained by SMNN(Competitive). This illustrates the better performance of ANNs against PM method in modeling EC-measured evaporation flux as a function of climatic data alone.

2.7.2.1 Identification of Optimal Combination of Input Variables

Since this case study represents a MISO process, the study is further extended to find the optimal combination of inputs that can characterize the evaporation process effectively. Also, the process of evapotranspiration is controlled by different factors at

different scales – vapor pressure deficit and stomatal processes at the scale of single leaf or tree, radiation as the driving variable at a regional scale (Jarvis and McNaughton, 1986). Different combinations of inputs were tested with the objective of minimizing the cost function shown in Equation (2.3). The results indicate that the use of net radiation and ground temperature alone as inputs to neural network models can result in better prediction accuracy. Although most of the evaporation models use a water vapor pressure gradient to estimate evaporation, inclusion of RH as one of the inputs to the neural networks model does not improve the performance of the model as RH is somewhat a redundant variable for the ANN model as the ANN model has already learnt the signal of RH which is embedded in the signal of GT, due to strong land-atmosphere interaction. This reiterates the findings of Lakshmi and Susskind (2001) and Wang et al. (2004), where it is reported that evaporation is not sensitive to atmospheric humidity since the land surface states contain the signals of near-surface atmospheric conditions as a result of strong land-atmosphere interaction. Inclusion of WS as one of the input variables to the neural networks model does not improve the performance of the model. Compared to neural network models using AT and NR as inputs, models using GT and NR resulted in better performance. As will be discussed later, variations in GT act in part as a surrogate variable to soil moisture and also have a longer memory of the feedback process inherent in the evaporation process. Also, Figure 2-3 reiterates that the best combination of inputs could be GT and NR as they have the greatest correlation with evaporation flux.

Table 2-5 presents the statistical performances of different neural network models using only net radiation and ground temperature as inputs. The optimal number of

neurons in the spiking layers of both SMNN(Competitive) and SMNN(SOM) is four. The architecture of the associator neural networks is ANN(2,4,1). The SMNNs performed better than the FF-NNs (Table 2-5), and in general, the use of net radiation and ground temperature alone as inputs resulted in an increase in the training error (Table 2-4, Table 2-5). However, during testing, better performance was obtained, indicating neural network models using net radiation and ground temperature alone as inputs have better generalization properties than do the neural network models utilizing all five inputs (air temperature, ground temperature, net radiation, relative humidity, and wind speed). The performance improvement in testing results and deterioration in training results is more apparent with regard to SMNNs, which is attributed to better generalization achieved due to the parallelization property of SMNNs.

The rate of evaporation is largely controlled by the energy and moisture available for evaporation. During "energy-limited" conditions, the energy balance at the land-atmosphere boundary layer determines the direction of movement of water vapor. Nevertheless, during "supply-limited" conditions, the water balance between the land and the atmosphere determines the rate of evaporation. Net radiation is the major factor influencing evaporation during energy limited conditions and soil moisture the most influential factor in determining evaporation during supply limited condition. Eltahir (1998) showed that an increase in soil moisture decreases the Bowen ratio, resulting in a decrease of ground temperature. By extension, variation in ground temperature can be considered as a surrogate variable for soil moisture due to the strong link between soil thermal properties and moisture status. This provides support for the optimal combination

of inputs (NR and GT) for the neural networks model. While NR accounts for energy-limited conditions, GT, as a surrogate, accounts for supply-limited conditions.

In order to demonstrate the modular learning of SMNNs, instances at which different spiking layer neurons spiked its corresponding associator neural networks is presented in Figure 2-7. The scatter plot on the left shows the variation of latent heat with respect to ground temperature and net radiation. The scatter plot on the right shows the mapping space associated with each associator neural network module. The mapping space of each associator neural network's module is represented by differently coloured points, which show that the SMNNs are effective in discretizing the complex mapping space into simpler domains that can be learned better. As mentioned before, clustering is carried out based on unsupervised learning (i.e. based on inputs alone). Points that are close to each other in the input space ideally should also be close to each other in the output space (i.e. points in one 2D-cluster based on the inputs should be in the same 3D-cluster based on the inputs and output). Figure 2-7 (3D-space) shows that there are few points (points that are in one cluster in the 2D-space and are not in the same cluster in the 3D-space) in certain regions of the input-output space. Those few points demonstrate that different combination of input variables (GT and NR) can result in similar output (LE).

2.7.2.2 Partitioning Analysis

Similar to the previous case study, partitioning analysis is carried out to assess the relative strengths of different models in predicting the evaporation flux above and below a certain threshold value. The testing dataset is considered for this analysis; it consists of 247 instances. The μ and the σ of the data are 87.12 W/m^2 and 85.36 W/m^2

respectively. A value slightly less than $\mu+\sigma$ is considered as the threshold (150 W/m^2) value. Out of the total 247 instances, 200 instances are below the threshold value and the remaining 47 instances are above the threshold value. Initially, partitioning analysis was carried out for models using AT, GT, NR, RH, and WS as inputs. The relative performance of these models above and below the threshold indicates that the SMNNs performed better than the FF-NNs in modeling values below the threshold (low evaporation flux) (Table 2-6). The performances of FF-NNs and SMNNs are comparable for the values above the threshold, indicating that SMNNs are more robust in capturing the dynamics of low evaporation flux. Partitioning analysis is also carried out for models using GT and NR alone as inputs to the models. Table 2-7 gives the performances of the models above and below the threshold value. Similar to the previous case, the SMNNs performed better than the FF-NNs in modeling the low evaporation flux and on par with the FF-NNs in modeling high evaporation flux.

2.8 Summary and Conclusions

In this study, a novel neural networks model called the spiking-modular neural networks (SMNNs) was proposed. Two variants of SMNNs were developed. The first variant, SMNN(Competitive), made use of a competitive layer as a spiking layer and the second variant, SMNN(SOM), made use of a self-organizing map as a spiking layer. The performance of the models was tested on two different case studies. The first case study involved modeling of streamflows and the second case study involved modeling of evaporation flux measured by a eddy-covariance (EC) system. While the first case study represented a single-input-single-output (SISO) process, the second case-study

represented a multiple-input-single-output (MISO) process. The rationale behind choosing these two case studies was to evaluate the performance of SMNNs on both simple and complex hydrological processes.

For the first case study (streamflow modeling), the SMNNs performed slightly better than the regular FF-NNs. Comparing SMNN(Competitive) and SMNN(SOM), the performance of the former model was better than that of the latter model. For both the SMNNs, the optimal number of neurons in the spiking layer was two, with the first neuron learning the dynamics of low flows and the second neuron learning the high flows. Partitioning analysis was carried out with respect to the performance of different models. It revealed that the performance of SMNNs is on par with that of FF-NNs for low flows. However, SMNNs perform better than the regular FF-NNs in modeling high flows.

For the second case study, initially the hourly latent heat flux was modeled as a function of air temperature (AT), ground temperature (GT), net-radiation (NR), relative humidity (RH), and wind speed (WS). The optimal number of clusters in the case of SMNNs was eight. The SMNNs were found to perform better than the FF-NNs in modeling evaporation flux. Results from the study revealed that different combinations of inputs may lead to similar values of evaporation flux, which indicates that the climatic variables are highly correlated with each other.

Since the second case study is a MISO process, the study was extended to find the optimal combination of input variables. Although most evaporation models use water-vapor pressure gradient to estimate evaporation, inclusion of RH as one of the inputs to the neural networks model, did not improve the performance of the neural networks model, reiterating the findings of Lakshmi and Susskind (2001) and Wang et al. (2004). For modeling EC-measured evaporation, the optimal combination of inputs was GT and NR. Partitioning analysis carried out to assess the relative strengths of different models in predicting the evaporation flux above and below a certain threshold value showed that SMNNs outperformed FF-NNs in modeling low-evaporation flux and on par with FF-NNs in modeling high-evaporation flux. It should be noted that the performance of the neural networks model depends on the data used for training the model. A neural networks model with good generalization ability is expected to perform better on sites similar to the one used for training the model. Hence, testing the robustness of the developed models on a nearby site may help in strengthening the results. Nonetheless, testing the robustness of the developed model on a completely different site may require re-training the model.

In general, for both case studies, SMNNs were found to perform better than FF-NNs. In the study, it is shown that SMNNs were successful in breaking down a complex mapping space into multiple relatively simpler mapping spaces that can be modeled with relative ease. The result from the study supports the findings of Zhang and Govindaraju (2000). As mentioned previously, the main difference between SMNN (Competitive) and SMNN (SOM) is that the former model makes use of a competitive layer as the spiking

layer, while the latter model makes use of SOM as the spiking layer. Functionally, SMNN(Competitive) learns the distribution of input space alone and SMNN(SOM) learns both the distribution and the topology of the input space. Since SMNN(Competitive) performed better than SMNN(SOM), it can be concluded that topology learning does not improve the performance of the SMNN model. This is due to the fact that, since individual neural network models are constructed for each cluster, the topology learned during the classification process does not influence the performances of associator neural networks.

The findings reported in this study are preliminary in nature and are based on two different case-studies. In order to verify and strengthen the findings of this research, the models have to be tested on further different case studies. Global optimization techniques such as genetic algorithms (GAs) are reported to be more robust than the conventional back-propagation (BP) algorithm in estimating the optimal values of weights and biases of neural networks. Hence the performance of SMNNs can further be improved by using GAs to train the associator neural networks. The proposed SMNNs are computationally intensive since they involve clustering of data and finding the optimal weights and biases of each associator neural networks. However, once trained, compared to regular FF-NNs, the SMNNs can be used with relative ease to accurately predict the hydrological variable of interest. The study reported in this chapter is a step in the direction to develop multiple local models rather than a single global model for hydrological processes.

2.9 Acknowledgements

The authors acknowledge the financial support of the Natural Sciences and Engineering Research Council (NSERC) of Canada through its Discovery Grants Program and the University of Saskatchewan through the Departmental Scholarship Program. The authors thank the Associate Editor (Dr. Steven Margulis) and three anonymous reviewers, whose comments greatly improved the quality of the paper.

2.10 References

Abbott, M. B., Bathurst, J. C., Cunje, J. A., O'Connell, P. E., and Rasmussen, J. (1986). "An introduction to the European hydrological system – System hydrologique Europeen, 'SHE', 1: History and philosophy of a physically-based, distributed modeling system." *J. Hydrol.*, 87, 45-59.

Arbib, M. A. (2003). *The handbook of brain theory and neural networks*, MIT Press, Cambridge, Mass.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. (2000a). "Artificial neural networks in hydrology. I: Preliminary concepts." *J. Hydrol. Eng.*, *5*(2), 115-123.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. (2000b). "Artificial neural networks in hydrology. II: Hydrologic applications." *J. Hydrol. Eng.*, *5*(2), 124-137.

Baker, J.M., and Griffis, T.J. (2005). "Examining strategies to improve the carbon balance of corn/soybean agriculture using eddy covariance and mass balance techniques." *Agric. Forest Meteorol., 128*, 163-177.

Blaney, H. F., and Criddle, W. D. (1950). "Determining water requirements in irrigated area from climatological irrigation data." *Soil Conservation Service Technical Paper No. 96, U.S. Department of Agriculture*, Washington DC, 48.

Bowden, G. J., Dandy, G. C., and Maier, H. R. (2005). "Input determination for neural network models in water resources applications. Part 1 – background and methodology." *J. Hydrol.*, 301, 75-92.

Brown, M., and Harris, C. (1994). *Neurofuzzy adaptive modeling and control*, Prentice Hall: New York.

Brutsart, W.H. (1982). *Evaporation into the Atmosphere*, Reidel Pub. Co., Boston.

Cai, S., Toral, H., Qiu, J., and Archer, J. S. (1994). "Neural network based objective flow regime identification in air-water two phase flow." *The Canadian Journal of Chemical Engineering,* 72, 440-445.

Demuth, H., and Beale, M. (2001). *Neural network toolbox learning. For use with MATLAB*. The Math Works Inc, Natick, Mass.

Drexler, J. Z., Snyder, R. L., Spano, D., and Paw, K. T. (2004). "A review of models and micrometeorological methods used to estimate wetland evapotranspiration." *Hydrol. Processes*, 18, 2071-2101.

Eltahir, E.A.B. (1998). "A soil moisture-rainfall feedback mechanism, 1: Theory and observations." *Water Resour. Res*., *34*(4), 765-776.

Entekhabi, D., Rodriguez-Iturbe, I., and Castelli, F. (1996). "Mutual interaction of soil moisture state and atmospheric processes." *J. Hydrol.*, 184, 3-17.

Hargreaves, G. H., and Samani, Z. A. (1982). "Estimating potential evapotranspiration." *J. Irrig. Drain. Eng.*, 108(3): 225-230.

Haykin, S. (1999). *Neural networks: A comprehensive foundation*, 2nd ed. MacMillan, New York.

Henderson-Sellers, A., Irannejad, P., McGuffie, K., and Pitman, A. (2003). "Predicting land-surface climates: Better skills or moving targets?" *Geophys. Res. Lett.*, 30(14), 1777.

Holdridge, L. R. (1962). "The determination of atmospheric water movements." *Ecology,* 43, 1-9.

Hong, Y., K. Hsu, S. Sorooshian, and X. Gao. (2005). "Self-organizing nonlinear output (SONO): A neural network suitable for cloud patch-based rainfall estimation at small scales." *Water Resour. Res.*, *41*, W03008, doi:10.1029/2004WR003142.

Hsu, K., Gupta, H. V., Gao, X., Sorooshian, S., and Imam, B. (2002). "Self-organizing linear output (SOLO): An artificial neural network suitable for hydrologic modeling and analysis." *Water Resour. Res.*, *38*(12), 1302, doi:10.1029/2001WR000795.

Jackson, R. D., Idso, S. B., and Reginato, R. J. (1976). "Calculation of evaporation rates during the transition from energy-limiting to soil-limiting phases using albedo data." *Water Resour. Res.*, 12(1), 23-26.

Jarvis, P. G., and McNaughton, K. G. (1986). "Stomatal control of transpiration: Scaling up from leaf to region." *Advances in Ecological Research.*, 15, 1-49.

Jensen, K. H. (1981). Unsaturated flow and evapotranspiration modeling as a component of the European hydrologic system (SHE), in *Modeling Components of Hydrologic Cycle*, edited by V. P. Singh, Water Resources Publications, Littleton, Col.

Karunanithi, N., Grenney, W. J., Whitley, D., and Bovee, K. (1994). "Neural networks for river flow prediction." *J. Comp. Civ. Eng.*, ASCE, *8*(2), 201-220.

Kohonen, T. (1989). *Self-organization and associative memory*, Springer-Verlag, New York.

Kumar, M., Raghuwanshi, N. S., Singh, R., Wallender, W. W., and Pruitt, W. O. (2002). "Estimating evapotranspiration using artificial neural network." *J. Irrig. Drain. Eng.*, 128(4), 224-233.

Lakshmi, V., and Susskind, J. (2001). "Utilization of satellite data in land-surface hydrology: Sensitivity and assimilation." *Hydrol. Processes*, 15(5), 877-892.

Leuning, R., and Judd, M.J. (1996). "The relative merits of open- and closed-path analysers for measurements of eddy fluxes." *Global Change Biology,* 2, 241-253.

Linacre, E. T. (1977). "A simple formula for estimating evaporation rates in various climates, using temperature data alone." *Agric. Meteorol.,* 18, 409-424.

MacKay, D. J. C. (1992). "Bayesian methods for adaptive models." Ph.D. Thesis, California Institute of Technology.

Maier, H., and Dandy, G. (2000). "Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications." *Environ. Modell. Software*, 15(1), 101-124.

Minns, A. W., and Hall, M. J. (1996). "Artificial neural networks as rainfall runoff models." *Hydrol. Sci. J.*, 41(3), 399-417.

Monteith, J. L. (1965). "Evaporation and environment in the state and movement of water in living organisms." *Society of Experimental Biology, Symposium No. 19*, Cambridge University Press, Cambridge, 205-234.

Penman, H. L. (1948). "Natural evaporation from open water, bare soil and grass." *Proceedings of the Royal Society, London*, 193, 120-146.

Priestley, C. H. B., and Taylor, R. J. (1972). "On the assessment of surface heat flux and evaporation using large scale parameters." *Mon. Weather Rev.*, 100, 81-92.

Sajikumar, N., and Thandaveswara, B. S. (1999). "A non-linear rainfall-runoff model using an artificial neural network." *J. Hydrol.*, 216, 32-55.

Salvucci, G. D. (1997). "Soil and moisture independent estimation of stage-two evaporation from potential evaporation and albedo or surface temperature." *Water Resour. Res.*, 33(1), 111-122.

Saxton, K. E. (1981). Mathematical modeling of evapotranspiration on agricultural watersheds, *Modeling Components of Hydrologic Cycle*, edited by V. P. Singh., pp. 183-204, Water Resources Publications, Littleton, Col.

Schotanus, P., Niewstadt, F.T.M., and De Bruin, H.A.R. (1983). "Temperature measurement with a sonic anemometer and its application to heat and moisture fluxes." *Bound. Lay. Meteorol.,* 26, 81-95.

Singh, V. P. (1989). *Hydrologic systems: Watershed modeling,* vol. II, Prentice-Hall, NJ.

Stephens, J. C., and Stewart, E. H. (1963). A comparison of procedures for computing evaporation and evapotranspiration, *Publication 62, International Association of Scientific Hydrology*, International Union of Geodynamics and Geophysics, Berkeley, CA, 123-133.

Sudheer, K. P., Gosain, A. K., and Ramasastri, K. P. (2003). "Estimating actual evapotranspiration from limited climatic data using neural computing technique." *J. Irrig. Drain. Eng.*, 129(3), 214-218.

Sudheer, K. P., Gosain, A. K., Rangan, D. M., and Saheb, S. M. (2002). "Modelling evaporation using an artificial neural network algorithm." *Hydrol. Processes*, 16, 3189-3202.

Temesgen, B., Eching, S., Davidoff, B., and Frame, K. (2005). "Comparison of some reference evapotranspiration equations for California." *J. Irrig. Drain. Eng.*, 131(1), 73-84.

Thornthwaite, C. W. (1948). "An approach toward a rational classification of climate." *Geog. Rev.*, 33, 55-94.

Tokar, A. S., and Markus, M. (2000). "Precipitation runoff modeling using artificial neural network and conceptual models." *J. Hydrol. Eng.*, 5(2), 151-161.

Trajkovic, S., Todorovic, B., and Stankovic, M. (2003). "Forecasting of reference evapotranspiration by artificial neural networks." *J. Irrig. Drain. Eng.*, 129(6), 454-457.

Twine, T. E., Kustas, W. P., Norman, J. M., Cook, D. R., Houser, P. R., Meyers, T. P., Prueger, J. H., Starks, P. J., and Wesely, M. L. (2000). "Correcting eddy-covariance flux underestimates over a grassland." *Agric. Forest Meteorol.*, 103, 279-300.

Wang, J., Salvucci, G. D., Bras, R. L. (2004). "An extremum principle of evaporation." *Water Resour. Res.*, 40, W09303, doi:10.1029/2004WR003087.

Webb, E.K., Pearman, G.I., Leuning, R. (1980). "Correction of flux measurements for density effects due to heat and water vapour transfer." *Q. J. R. Meteorol. Soc.,* 106, 85-100.

Zhang, B., and Govindaraju, S. (2000). "Prediction of watershed runoff using Bayesian concepts and modular neural networks." *Water Resour. Res.*, 36(3), 753-762.

Table 2-1 Statistical Properties of Streamflow Data between Umfreville and Sioux Lookout

Statistics	Entire Data Set		Training Data Set		Testing Data Set	
	Umfreville	Sioux Lookout	Umfreville	Sioux Lookout	Umfreville	Sioux Lookout
Minimum, m^3/s	3.2	16.2	3.2	18.5	9.7	16.2
Maximum, m^3/s	372.0	634.0	372.0	634.0	274.0	511.0
Median, m^3/s	41.3	90.6	39.5	88.4	47.3	93.8
Average, m^3/s	57.0	121.0	55.0	120.7	62.1	121.7
SD[a], m^3/s	45.5	87.3	44.6	87.7	47.4	86.3
CV[b]	0.8	0.7	0.8	0.7	0.8	0.7

[a]SD is standard deviation
[b]CV is coefficient of variation

Table 2-2 Statistical Performance of Different Models in Modeling Streamflows[a]

Model	Training			Testing		
	RMSE, m^3/s	MRE	R	RMSE, m^3/s	MRE	R
FFNN	27.9	0.17	0.95	30.4	0.22	0.96
SMNN (Competitive)	28.4	0.17	0.95	27.5	0.22	0.96
SMNN(SOM)	28.8	0.17	0.94	28.4	0.22	0.96

[a]RMSE is root-mean-square error; MRE is mean relative error; FFNN is feed forward neural network; SMNN is spiking modular neural network; SOM is self-organizing map.

Table 2-3 RMSE and MRE Statistics of Different Models Above ans Below the Threshold Streamflow Modeling[a]

Model	Flow Rate < 200 m3/s		Flow Rate > 200 m3/s	
	RMSE, m^3/s	MRE	RMSE, m^3/s	MRE
FFNN	24.2	0.24	53.5	0.13
SMNN (Competitive)	24.1	0.24	41.9	0.12
SMNN (SOM)	25.4	0.25	41.4	0.12

[a]Abbrevations as in Table 2-2

Table 2-4 Statistical Performance of Different Models in Modeling Evaporation[a]

Model	Training			Testing		
	RMSE, W/m^2	MRE	R	RMSE, W/m^2	MRE	R
FFNN	37.0	2.4	0.90	73.4	1.6	0.69
SMNN (Competitive)	32.2	2.2	0.93	70.2	1.2	0.71
SMNN(SOM)	33.4	2.0	0.92	73.0	1.3	0.67

[a] Abbrevations as in Table 2-2

Table 2-5 Statistical Performance of Different Models in Modeling Evaporation With Net Radiation and Ground Temperature Alone as Inputs[a]

Model	Training			Testing		
	RMSE, W/m^2	MRE	R	RMSE, W/m^2	MRE	R
FFNN	43.7	3.0	0.86	67.2	1.5	0.72
SMNN (Competitive)	43.9	3.3	0.86	64.4	1.1	0.74
SMNN(SOM)	43.9	3.5	0.86	65.9	0.9	0.73

[a] Abbrevations as in Table 2-2

Table 2-6 RMSE and MRE Statistics of Different Models Above and Below the Threshold When Air Temperature, Ground Temperature, Net Radiation, Relative Humidity, and Wind Speed Are Considered as Inputs[a]

Model	Evaporation Flux < 150 W/m^2		Evaporation Flux > 150 W/m^2	
	RMSE, W/m^2	MRE	RMSE, W/m^2	MRE
FFNN	61.3	1.8	111.1	0.4
SMNN (Competitive)	57.4	1.4	109.2	0.3
SMNN (SOM)	61.3	1.6	109.5	0.4

[a]Abbrevations as in Table 2-2

Table 2-7 RMSE and MRE Statistics of Different Models Above and Below the Threshold When Ground Temperature and Net Radiation Are Considered as Inputs[a]

Model	Evaporation Flux < 150 W/m^2		Evaporation Flux > 150 W/m^2	
	RMSE, W/m^2	MRE	RMSE, W/m^2	MRE
FFNN	56.6	1.8	100.4	0.3
SMNN (Competitive)	53.5	1.2	98.2	0.3
SMNN (SOM)	55.0	1.0	99.6	0.3

[a]Abbrevations as in Table 2-2

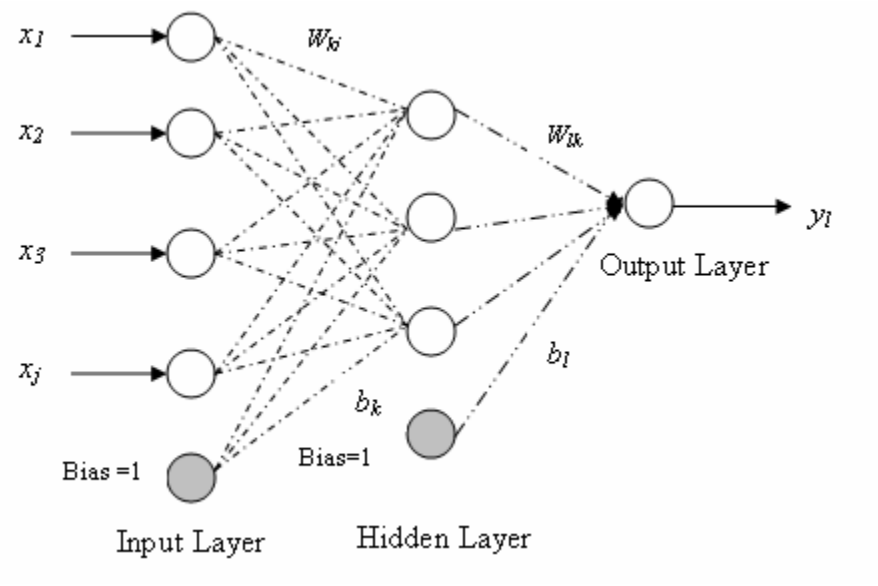

Figure 2-1 Structure of the three-layered feed forward neural network (FFNN).

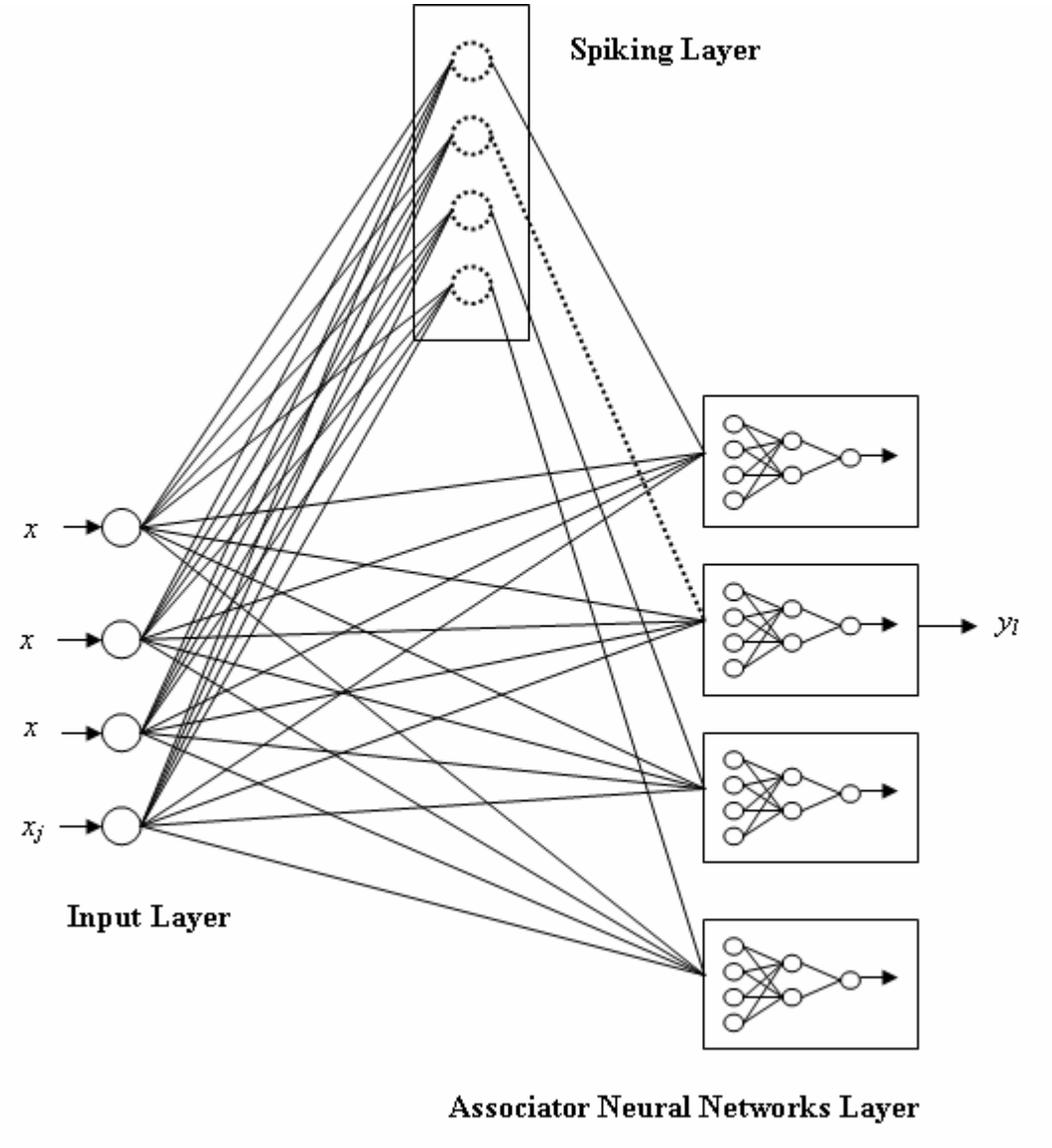

Figure 2-2 Structure of the Spiking Modular Neural Network (SMNN)

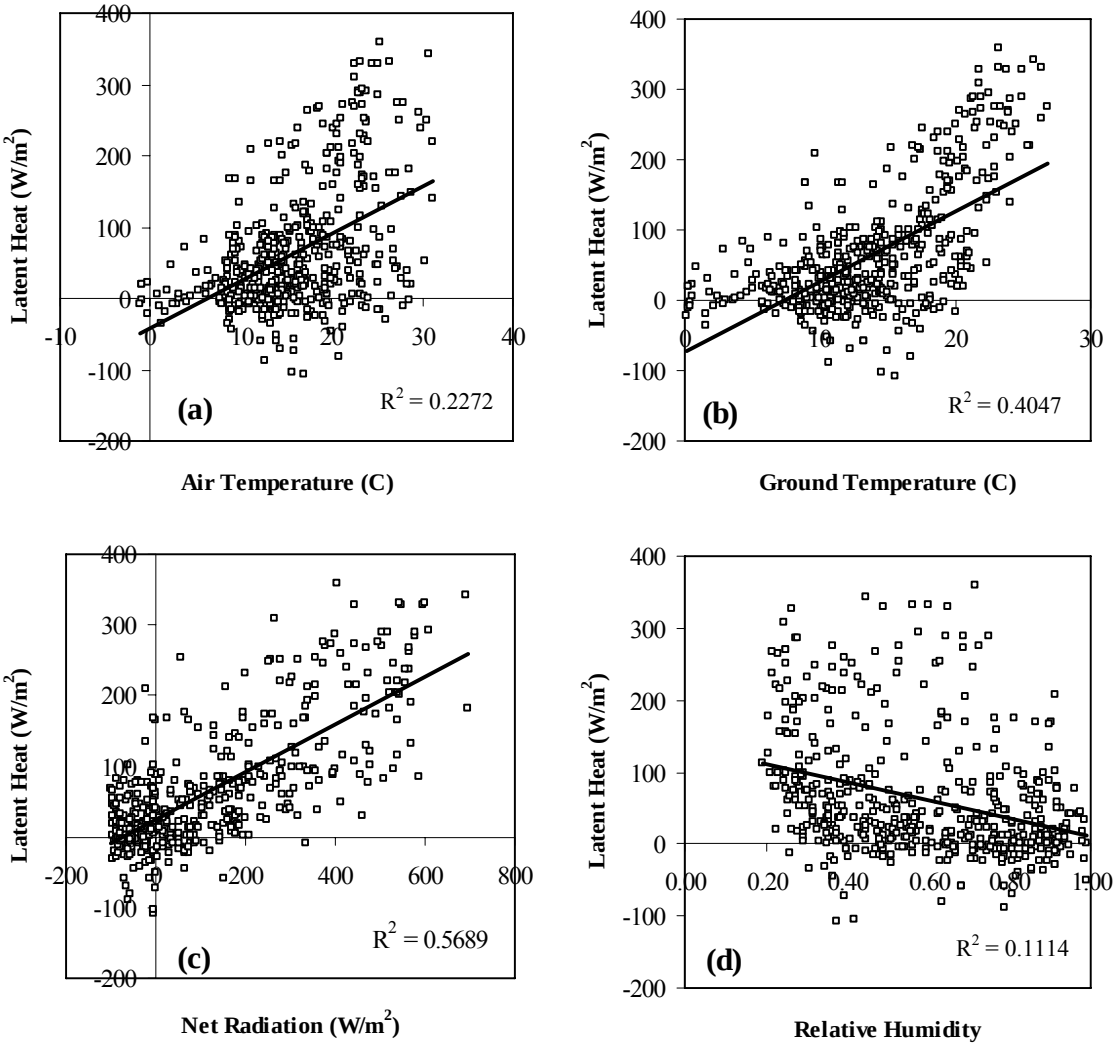

Figure 2-3 Correlation plot between latent heat and (a) air temperature, (b) ground temperature, (c) net radiation, and (d) relative humidity

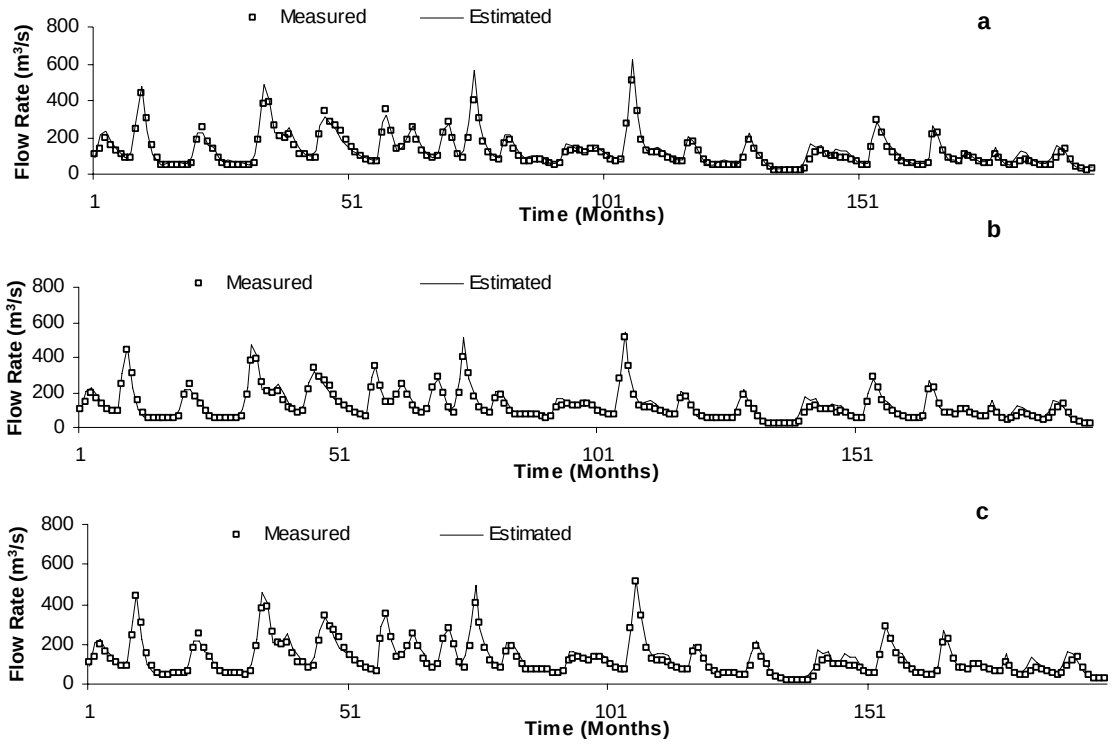

Figure 2-4 Comparison of measured and estimated flows by (a) FFNNs, (b) SMNN(Competitive), and SMNN(SOM)

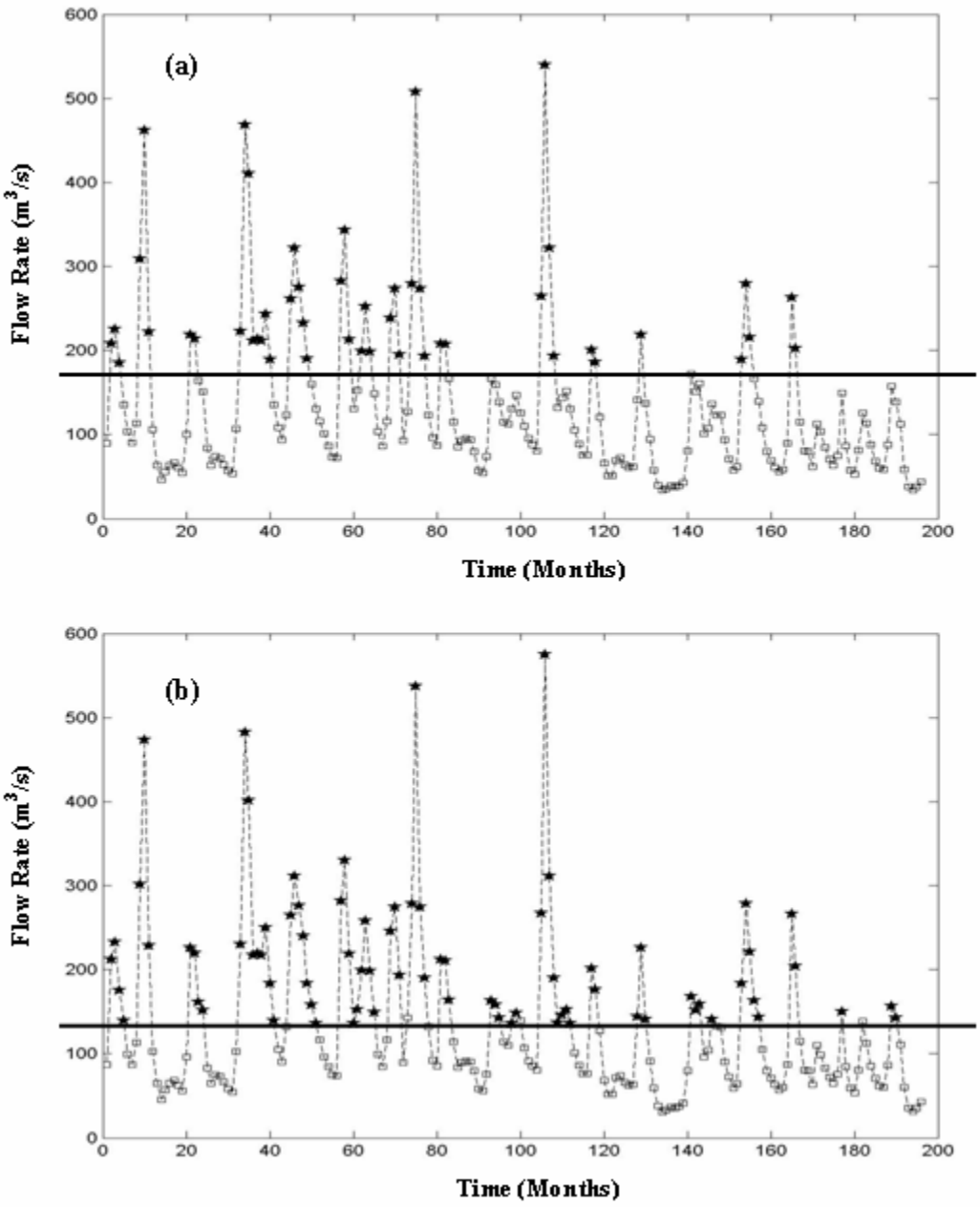

Figure 2-5 Plots showing the instances at which different spiking layer neurons fired: (a) SMNN(Competitive) and (b) SMNN(SOM). Solid lines indicate threshold value, stars indicate instances at which spiking layer neuron 1 fired, and open rectangles indicate instances at which spiking layer neuron 2 fired.

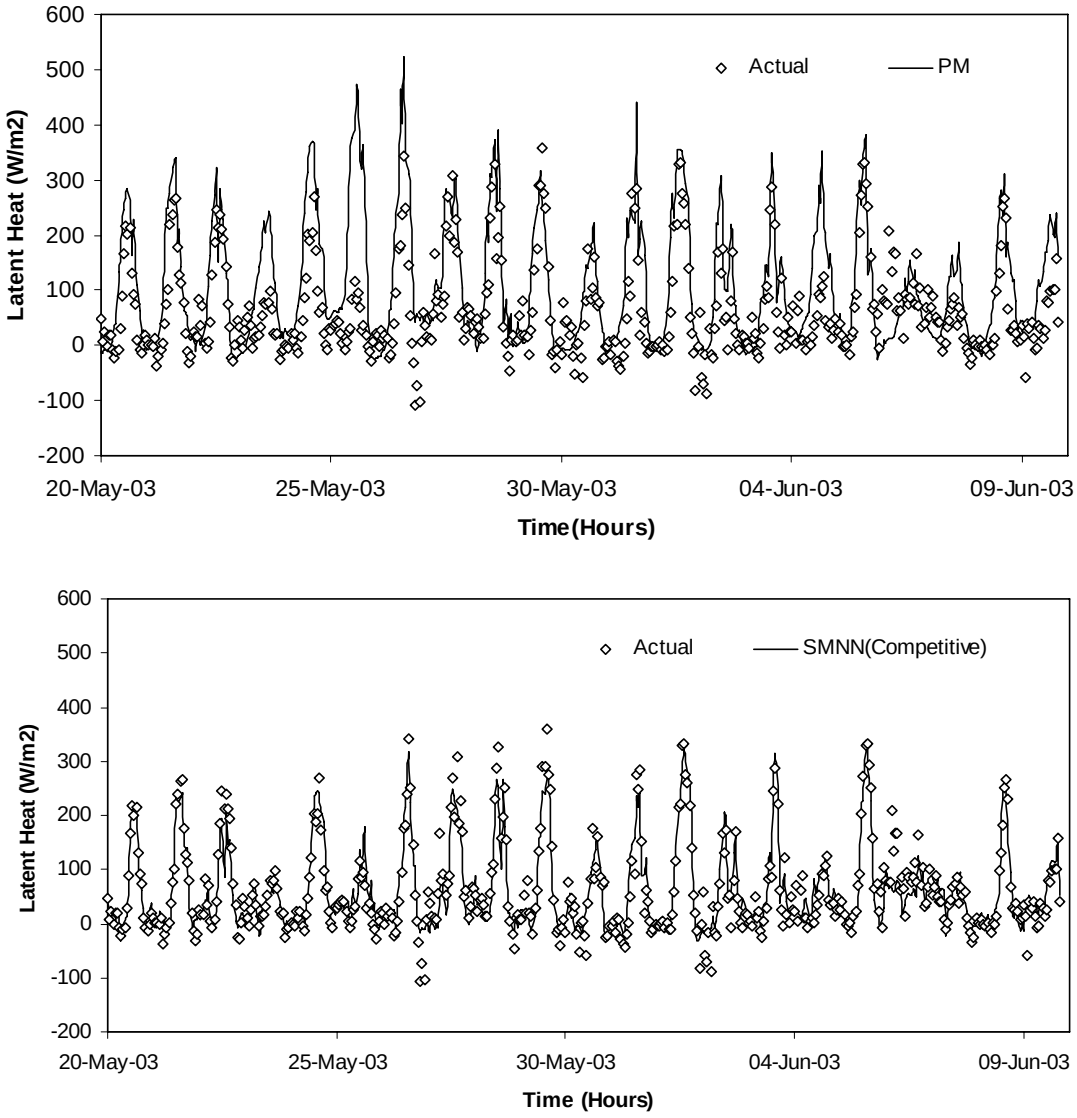

Figure 2-6 Comparison of measured evaporation flux with (a) Penman-Monteith and (b) SMNN(Competitive) estimates.

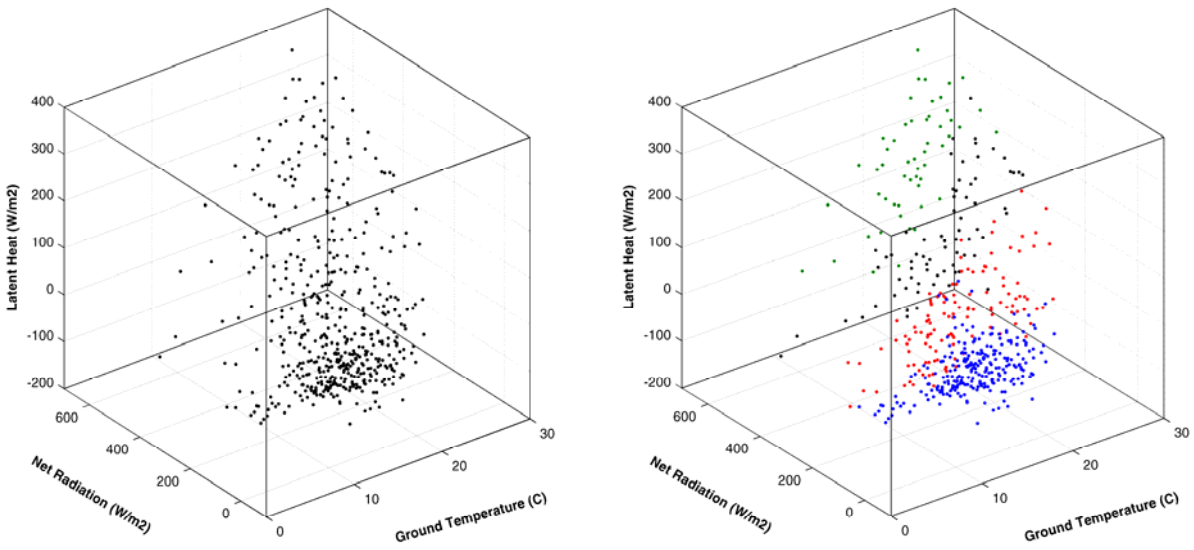

Figure 2-7 Scatterplots illustrating the performance of SMNN.

Chapter 3 - Modelling the Dynamics of the Evapotranspiration Process Using Genetic Programming

Citation: Parasuraman, K., Elshorbagy, A., and Carey, S. K. (2007). "Modelling the dynamics of the evapotranspiration process using genetic programming." *Hydrol. Sci. J.,* 52(3), 563-578.

Contribution of the PhD candidate

Model conceptualization was carried out by Kamban Parasuraman and Dr. Amin Elshorbagy. Computer program development and simulations were carried out by Kamban Parasuraman. The text of the published paper was created by Kamban Parasuraman, with Dr. Amin Elshorbagy critically reviewing the manuscript. Dr. Sean Carey carried out the preliminary raw data quality checks, provided the site description details, and offered editorial guidance.

Contribution of this chapter to the overall study

Similar to the previous chapter, this study was also aimed at improving the utility of the data-driven models beyond forecast applications as a tool for scientific investigations. Nevertheless, the ability of another promising data-driven technique, namely, genetic programming (GP), is evaluated in this regard. The hypothesis is that, the robustness of GP to evolve its own model structure with relevant parameters could aid in understanding and improving our knowledge of the predictand-predictor relationship. The

hypothesis was tested by applying GP to model the dynamics of the evapotranspiration process from two case-studies with different morphological characteristics. The ability of GP to arrive at an explicit model structure for modeling evapotranspiration process is demonstrated in this study, and from the insights gained by analyzing the GP-evolved equations, it was concluded that the net-radiation and ground-temperature are the most important state variables for characterizing the actual evapotranspiration process.

3.1 Abstract

Evapotranspiration constitutes one of the major components of the hydrological cycle and hence its accurate estimation is of vital importance to assess water availability and requirements. This study explores the utility of genetic programming (GP) to model the evapotranspiration process. An important characteristic of GP is that both the model structure and coefficients are simultaneously optimized. The method is applied in modelling eddy-covariance (EC)-measured latent heat (LE) as a function of net radiation (NR), ground temperature (GT), air temperature (AT), wind speed (WS) and relative humidity (RH). Two case studies having different climatic and topographic conditions are considered. The performance of the GP model is compared with artificial neural network (ANN) models and the traditional Penman-Monteith (PM) method. Results from the study indicate that both the data-driven models, GP and ANNs, performed better than the PM method. However, performance of the GP model is comparable with that of the ANN model. The GP-evolved models are dominated by NR and GT, indicating that these two inputs can represent most of the variance in LE. The results show that the GP-evolved equations are parsimonious and understandable, and are well suited to modelling the dynamics of the evapotranspiration process.

3.2 Introduction

Approximately 75% of the total annual precipitation on land surfaces is returned back to the atmosphere in the form of evaporation and transpiration (Singh, 1989), illustrating their importance in the global water balance. Due to complex interactions

between the land–plant–atmosphere systems, evapotranspiration (the term used to collectively describe both evaporation and transpiration) remains poorly characterized for many surfaces (Souch et al., 1996). Moreover, the evapotranspiration process is embedded with large variability in both spatial and temporal scales. Unlike precipitation and river flow, which can be directly measured, evapotranspiration is usually estimated based on mass transfer, energy transfer, or water budget methods. Traditional measurements of evaporation by pan-evaporimeter and lysimeter are subject to a large set of assumptions, cumbersome and labour-intensive, and may not be appropriate for large-scale studies. More recently, micrometeorological methods such as energy-balance–Bowen-ratio (EBBR) and eddy-covariance (EC) have found widespread application to measure actual evaporation and improved our understanding of the evaporation process (Drexler et al., 2004).

Numerous attempts have been made to model evaporation and/or evapotranspiration based on climatological data. Important historical examples include: (a) empirical relationships between meteorological variables (Blaney and Criddle, 1950; Stephens and Stewart, 1963; Priestley and Taylor, 1972) and (b) physically-based equations (Penman, 1948; Monteith, 1965). While the former methods estimate evaporation based on climate data, the latter methods link evaporation dynamics with the supply of energy and the aerodynamics transport characteristics of a natural surface.

Evapotranspiration models are based on different conceptual rates, such as potential, actual, wet environment, and reference crop evapotranspiration rates. For

systems that are poorly understood, it is difficult to choose an appropriate model to estimate actual evapotranspiration as vegetation, climate and water availability vary widely in space and time and strongly influence the evapotranspiration process. Almost all models of evapotranspiration reflect some measure of meteorological control over the evaporative processes (i.e. potential evapotranspiration). The complexity of the model varies depending upon the uniformity of the surface, canopy properties, and baseline assumptions (see Shuttleworth and Wallace 1985; Choudhury and Monteith 1988; Granger and Gray 1989; Flerchinger et al. 1996; Biftu and Gan, 2000). Among the wealth of local and global evapotranspiration models, the Penman-Monteith (PM) (Monteith, 1965) equation is perhaps the most widely adopted evapotranspiration model (Abbott et al., 1986). The PM method estimates reference evapotranspiration for a hypothetical uniform reference grass surface fulfilling certain requirements. Evapotranspiration rate can then be obtained by multiplying the reference evapotranspiration by the crop coefficient. The PM method is shown to perform well for dense, closed canopy situations and for other wet vegetated surfaces (Shuttleworth, 1991). However, the applicability of the PM equation requires surface and aerodynamic resistance data, which are not readily available. Also, it should be noted that the evapotranspiration calculated by the PM method is potential evapotranspiration (unlimited supply of soil water) for a well-watered surface and not actual evapotranspiration (water limited). The potential estimates of evapotranspiration given by the PM equation can be converted to actual evapotranspiration by considering the soil moisture limitations. However, this conversion is cumbersome and involves large uncertainty due to the following reason: compared to climatic data, soil moisture data are not always readily available at the temporal

resolution of other climatic variables. Hence, in addition to the large variability of soil moisture at both spatial and temporal scales, the uncertainties instigated by the interpolation of soil moisture data to match the temporal resolution of climatic variables also occur.

The complexities inherent in modelling evapotranspiration using conceptual models provide impetus to test the utility of data-driven models. Unlike conceptual models, data-driven models do not emphasize the nature of the system and the physical laws governing the system. For example, to forecast reference crop evaporation, Tracy et al. (1992) used simple yearly differencing or monthly average models, Mariño et al. (1993) adopted a seasonal autoregressive integrated moving average (SARIMA) model, and Hameed et al. (1995) investigated the utility of a transfer-function noise model. However, these data-driven models are based on linear systems theory and hence may not be well suited for characterizing a nonlinear process such as evapotranspiration. Furthermore, the application of these traditional deductive data-driven models to represent any process requires a prior definition of model structure. Nevertheless, difficulties associated with model structure identification in the case of the complex evapotranspiration process impede the utility of these approaches in modelling the above process. Recently, artificial neural networks (ANNs) have been adopted as an alternative inductive data-driven modelling tool for modelling evapotranspiration. The ability of ANNs to identify and learn the input–output patterns without being explicitly programmed to do so, makes them a promising tool to model complex hydrological processes. Key examples of the application of ANNs in hydrology includes: rainfall–

runoff modelling (Hsu et al., 1995; Minns and Hall, 1996; Shamseldin, 1997), and rainfall forecasting (French et al., 1992; Zhang et al., 1997). More information on the application of ANNs in water related studies can be found in the ASCE Task Committee on the Application of Artificial Neural Networks in Hydrology (2000) and in Maier and Dandy (2000). Buoyed by the success of ANNs in modelling complex hydrological processes, a limited number of studies have been undertaken to model the dynamic evaporation and/or evapotranspiration process using ANNs. This includes studies by Kumar et al. (2002), Sudheer et al. (2003), Trajkovic et al. (2003) and Parasuraman et al. (2006). While Kumar et al. (2002) and Trajkovic et al. (2003) modelled the PM estimates of evaporation, the study by Sudheer et al. (2003) considered lysimeter-measured actual evaporation for modelling purposes. In Parasuraman et al. (2006), the first attempt of modelling the EC-measured actual evapotranspiration was made.

Another promising inductive data-driven technique is genetic programming (GP) introduced by Koza (1992), is a method for constructing populations of models using stochastic search methods, namely evolutionary algorithms. An important characteristic of GP is that both the variables and constants of the candidate models are optimized. Hence, compared to other regression techniques, it is not required to choose the model structure a priori. In water-related studies, GP has been applied to model: flow over a flexible bed (Babovic and Abbott, 1997), the rainfall–runoff process (Whigham and Crapper, 2001; Savic et al., 1999), runoff forecasting (Khu et al., 2001), urban fractured-rock aquifer dynamics (Hong and Rosen, 2002), temperature downscaling (Coulibaly, 2004), and the rainfall-recharge process (Hong et al., 2005).

Although GP and ANNs can be seen as alternative techniques for the same task, such as, e.g. classification and approximation problems, in contrast to ANNs, GP has not been used extensively for modelling hydrological processes. The robustness of GP in modelling complex nonlinear processes warrants its application in modelling the actual evapotranspiration process, which is embedded with nonlinearity in both spatial and temporal scales. To the knowledge of the authors, no work has been reported in the literature on modelling actual evapotranspiration using GP. Hence in this study, an attempt has been made to evaluate the ability of GP in modelling EC-measured actual evapotranspiration. Specific objectives of the study include modelling the EC-measured latent heat flux (LE) (the product of the latent heat of vaporization and evapotranspiration) using GP for two distinct case studies, and comparing its performance with the PM estimates and the ANN model. The resulting GP-evolved models are analysed to understand and improve our knowledge of the predictand–predictor relationship.

3.3 Materials and Methods

3.3.1 Artificial Neural Networks

Artificial neural networks (ANNs) are essentially a semi-parametric regression technique with the ability to approximate any measurable function up to an arbitrary degree of accuracy. According to Haykin (1999), ANNs are a massively parallel distributed information processing system that is capable of storing the experiential

knowledge gained by the process of learning, and of making it available for future use. Feed-forward neural networks (FF-NNs) are the most widely adopted network architecture for the prediction and forecasting of water resource variables (Maier and Dandy, 2000). Typically, FF-NNs consist of an input layer, hidden layer(s) and an output layer. The input layer is connected to the hidden layer and in turn the hidden layer is connected to the output layer by means of connection weights. The hidden layer neurons consist of activation functions which help in translating the input variables to the required output variables.

In this study, a regular three layered FF-NN with J input neurons, K hidden neurons, and L output neurons is considered. Symbolically, the above ANN architecture can be represented as $ANN(J,K,L)$. Let j, k, and l be the indices representing the input, hidden, and output layers respectively. The FF-NN makes use of the tan-sigmoidal activation function in the hidden layer and the linear activation function in the output layer. The transformation of inputs ($x_1, ..., x_j$) to output (y_l) is achieved by Equations (3.1) and (3.2):

$$y_l = f_1\left[\sum_{k=1}^{K} w_{lk} f_2\left(\sum_{j=1}^{J} w_{kj} x_j + b_k\right) + b_l\right] \tag{3.1}$$

$$f_2(p) = \frac{2}{\left(1 + e^{-2p}\right)} - 1 \tag{3.2}$$

where w_{kj} represents the connection weight between the jth input neuron and kth hidden neuron, and w_{lk} represents the connection weight between the kth hidden neuron

and lth output neuron. Parameters b_k and b_l represent the bias of the corresponding hidden and output layer neurons. Function $f_1(\cdot)$ represents the linear activation function and $f_2(\cdot)$ represents the tan-sigmoidal activation function. While the tan-sigmoidal activation function squashes the input between –1 and +1, the linear activation function calculates the neuron's output by simply returning the value passed to it.

ANN modelling of a process demands two operations: training and testing. Training involves optimizing the connection weights through minimization of a certain cost function. In order to make the training process more efficient, both the inputs and output variables were normalized between the interval –1 and +1. Once the weights of the ANN model have been determined, it can be tested by evaluating its performance on a data set other than the training set, which is the testing set. The typical cost function used in training FF-NNs involves minimizing the mean sum of squares of the network errors (MSE). However in this study, in order to overcome the problem of over-fitting, the Bayesian-regularization back-propagation algorithm (Demuth and Beale, 2001) is used for training the FF-NNs. This algorithm improves the generalization property of the ANN model by developing networks with smaller weights and biases, and thus a smoother response that is less likely to result in over-fitting (Demuth and Beale, 2001). Hence, along with MSE, the cost function (Equation (3.3)) involves minimizing the mean of the sum of squares of the network bias and connection weights (MSW):

$$\mathrm{M}SE\_REG = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i'\right)^2 + (1-\alpha)\left(\frac{1}{N}\sum_{j=1}^{N} w_j^2\right) \tag{3.3}$$

where y_i and y_i' represent the measured and computed counterparts; α represents the regularization parameter; n and N represents the number of training instances and the number of network parameters, respectively. The regularization parameter α is determined in a Bayesian framework using automatic relevance determination, where different weight decays for each input and layer are set automatically. More information on automatic relevance determination can be found in (MacKay, 1992).

One of the important issues in the development of a neural network model is the determination of the optimal configuration of the neural network model. The optimal number of hidden neurons is usually determined by a trial and error method. However, the significant input variables for characterizing a process is usually determined by linear cross-correlation (Bowden et al., 2005). Linear cross-correlation can only detect linear dependence between two variables, and is not suited for capturing the nonlinear dependence between the inputs and the output. A review of different methods of input selection adopted in water resources literature can be found in Bowden et al. (2005). In this study, the optimal network architecture is determined by performing a systematic search of different network configuration and user-adjustable parameters, with the objective of minimizing the cost function.

3.3.2 Genetic Programming

Genetic programming (GP), introduced by Koza (1992), belongs to a class of evolutionary algorithms (EA), which are based on the concepts of natural selection and genetics. Genetic programming is a relatively new addition to a pool of other EA such as

evolutionary programming (Fogel et al., 1966), genetic algorithms (Holland, 1975) and evolution strategies (Schwefel, 1981). Genetic symbolic regression (GSR) (Koza, 1992) is a special application of GP in the area of symbolic regression, where the objective is to find a mathematical expression that fits the given pairs of values; GSR can be considered as an extension of numerical regression problems where, for a given set of values of various independent variables and the corresponding values of dependent variables, one predetermines the functional form (linear, quadratic, or polynomial) of the model. The objective is to find the set of numerical coefficients that best fits the model. However, GSR involves finding the mathematical expression in symbolic form (both the discovery of the optimal functional form and the appropriate numerical coefficients), which provides the optimal fit between a finite sample of independent variables and dependent variables. Hence, the purpose of GSR is to develop mathematical models between the predictand and the predictor variables.

Genetic symbolic regression works with two sets of variables, namely the functional set and terminal set (Koza, 1992). The terminal set consists of independent variables and constants, and the functional set consists of basic mathematical operators {+, –, *, /, sin, cosh, log, power …} that may be used to form the model. The choice of operators depends upon the degree of complexity of the problem to be modelled. Genetic symbolic regression works by constructing a population of mathematical models from different combinations of the functional and terminal sets. Each model (individual) in the population can be considered as a potential solution to the problem. The mathematical models are usually coded in a parse tree form. For example, Figure 3-1 shows the parse

tree notation of a mathematical model $f(x,y,z) = (5 + x)\times(y - z)$. In Figure 3-1, the connection points are called nodes, and it can be seen that the inner nodes of the parse tree are made up of functions and the terminal nodes are made up of variables and constants.

This section outlines the GP algorithm adopted in this study. For a detailed description of the GP method, the readers are referred to Koza (1992) and Babovic and Keijzer (2000). The first step in implementing GP is to generate the initial population for a given population size. This study adopts the ramped half-and-half method to initialize the population as it generates parse trees of various sizes and shapes and also provides a good coverage of the search space (Koza, 1992). Once initialized, the fitness of each individual (mathematical model) in the population is evaluated based on some objective function. Fitness is a numerical value attached to each individual based on their performance. The higher the fitness of an individual, the greater the chance of that individual being carried over to the next generation. At each generation, new sets of models are evolved by applying the genetic operators: selection, crossover and mutation (Koza, 1992; Babovic and Keijzer, 2000). These new models are called offspring and they form the basis for the next generation. In this study, the fitness measure is evaluated based on the root mean squared error (RMSE).

Once the fitness of individual models in the population is evaluated, the next step is to carry out selection. Selection can be carried out by several methods, such as truncation selection, tournament selection, fitness proportional selection and roulette

wheel selection (Koza, 1992). The latter method has been adopted in this study as it is straightforward to implement. The roulette wheel is constructed by proportioning the space in the roulette wheel based on the fitness of each model in the population. The selection process ensures that the models with higher fitness have more chance of being carried over to the next generation. The process of selection leads to the creation of a temporary population, called the mating pool. The models in the mating pool are acted upon by the genetic operators, crossover and mutation.

The role of the crossover operator is to generate new models, which did not exist in the old population, so that the problem space is sampled thoroughly. Crossover is carried out by choosing two parent models from the mating pool and swapping corresponding sub-tree structures across a randomly chosen point to produce two different offspring with different characteristics. The number of models undergoing crossover depends upon the chosen probability of crossover, P_c. Mutation involves random alteration of the parse tree at the branch or node level. This alteration is done based on a chosen probability of mutation, P_m. Mutation introduces new offspring into the population and thereby guards against premature convergence. Figure 3-2 demonstrates the crossover and mutation operators. The crossover point between Parent 1 and Parent 2 is shown by the dashed line and the corresponding sub-tree structures are swapped, resulting in Offspring 1 and Offspring 2. In Offspring 1, the terminal node has undergone mutation (2 replaced by *Z*). Thus, it can be seen that the genetic operators, crossover and mutation are able to produce new models (offspring) that are structurally different from their parent models. The various GP parameters adopted in this study are given in Table

3-1. In this study, the GP system used is an adaptation of GPLAB (Silva, 2005), a GP toolbox for MATLAB.

The basic steps involved in GP can be summarised as follows:

1. Identify the functional and terminal sets, along with the fitness measure.
2. Generate the initial population randomly from functional and terminal sets.
3. Based on the fitness measure, evaluate the fitness of each individual.
4. Apply the selection operator. The higher the fitness of an individual, the greater the chance of that individual being selected and carried over to the next generation (survival of the fittest). This temporary population is termed the mating pool.
5. Based on the probability of crossover (P_c), pairs of individuals from the mating pool are chosen and a crossover function is performed.
6. The next step is to apply a mutation operator based on the probability of mutation (P_m). Mutation helps in ensuring that no point in the individual search space remains unexplored.
7. Copy the resultant individuals to a new population.
8. Repeat steps 3 to 7 for a predetermined number of iterations or until a specified value of the cost function is reached.

3.3.3 Performance Evaluation

The performance of different models is evaluated based on multi-criteria analysis. The root mean squared error (RMSE), the mean absolute relative error (MARE),

and correlation coefficient (R) have been considered to carry out this analysis. Each of the above performance statistics provides different information about the predictive ability of the model. The RMSE statistic indicates only the model's ability to predict away from the mean (Hsu et al., 1995). The RMSE gives more weight to high values because it involves squaring the difference between observed and predicted values. The MARE provides an unbiased error estimate because it gives appropriate weight to all magnitudes of the predicted variable. The correlation statistic, *R*, evaluates the linear correlation between the measured and the computed values. RMSE, MARE and *R* are calculated using Equations (3.4), (3.5), and (3.6) respectively, where *N* represents the number of instances presented to the model; y_i and y_i' represent measured and computed counterparts; and $\bar{y}$ represents the mean of the corresponding variable.

$$RMSE = \left[\frac{1}{N} \sum_{i=1}^{N} (y_i - y_i')^2 \right]^{0.5} \tag{3.4}$$

$$MARE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i - y_i'}{y_i} \right| \tag{3.5}$$

$$R = \frac{\sum_{i=1}^{N} \left(y_i - \overline{y_i} \right) \left(y_i' - \overline{y_i'} \right)}{\sqrt{\sum_{i=1}^{N} \left(y_i - \overline{y_i} \right)^2 \sum_{i=1}^{N} \left(y_i' - \overline{y_i'} \right)^2}} \tag{3.6}$$

3.3.4 Case Studies

3.3.4.1 Case study I

For the first case study, EC-measured evapotranspiration from the South Bison Hill (SBH) (57°39′N; 111°13′W), a waste-rock overburden pile located at the Syncrude Canada Ltd, Mildred Lake mine north of Fort McMurray, Alberta, Canada, is considered. The SBH was constructed with waste-rock material from oil sands mining in stages between 1980 and 1996. The area of SBH is 2 km^2; it rises 60 m above the surrounding landscape and has a large flat top several hundred metres in diameter. To reclaim the overburden so that revegetation can occur, the underlying shale is covered by a 0.2 m layer of peat mineral mix on top of a 0.8 m layer of glacial till. The top of the SBH is dominated by foxtail barley (*Hordeum jubatum*); also present are other minor species such as fireweed (*Epilobium angustifolium*). In this case study, the hourly EC-measured LE flux between 20 May and 25 August 2003 is considered. However, for modelling purposes, the day-time (08:00–20:00) evapotranspiration alone is considered. Disregarding the missing values, the number of instances considered for training and testing are 658 and 381, respectively. The coefficient of variation of NR, AT, GT, RH, WS and LE is 0.64, 0.29, 0.28, 0.39, 0.49 and 0.72, during training, and 0.65, 0.24, 0.22, 0.34, 0.50 and 0.67 during testing, respectively.

3.3.4.2 Case study II

The evapotranspiration data from the South West Sand Storage (SWSS) facility, which is located several kilometres from SBH at the Mildred Lake mine, is considered in

the second case study. The SWSS is currently the largest operational tailings dam in the world, holding approximately 435 × 10^6 m^3 of material, covering 25 km^2, and standing approximately 40 m high with a 20H:1V side-slope ratio. Side-slopes are constructed as 100 m wide berms connected by 10% slopes to form an overall slope of 5%. Soils consist of mine tailings sand overlain with 0.4 to 0.8 m of topsoil that is a mixture of peat and mineral soil with a clay loam texture. Both vegetation species and composition vary across the SWSS, with dominant groundcover including horsetail (*Equisetum arvense*), fireweed (*Epilobium angustifolia*), sow thistle (*Sonchus arvense*), and white and yellow sweet clover (*Melilotus alba*, *Melilotus officinalis*). Tree and shrub species include Siberian larch (*Larix siberica*), hybrid poplar (*Populus* sp. hybrid), trembling aspen (*Populus tremuloides*), white spruce (*Picea glauca*) and willow (*Salix* sp.). The EC-measured LE flux at this site from 15 May to 10 September 2005 is considered. Similar to the previous case study, the day time evapotranspiration alone is considered for modelling purposes. Disregarding the missing values, the number of instances considered for training and testing purposes are 787 and 408 respectively. The coefficient of variation of NR, AT, GT, RH, WS and LE, during training is 0.60, 0.28, 0.20, 0.34, 0.46 and 0.49, respectively. The corresponding values during testing are 0.73, 0.27, 0.14, 0.27, 0.47 and 0.55.

Estimation of evaporation from these reconstructed watersheds is of vital importance as it plays a major role in the water-balance of the system, which links directly to ecosystem restoration strategies. Air temperature (AT; ºC), ground temperature (GT; ºC), net radiation (NR; W m^{-2}), relative humidity (RH), and wind speed

(WS; m s^{-1}) were measured by the weather station located on top of both the sites. Turbulent fluxes of heat and water vapour were measured using a CSAT3 sonic anemometer and thermometer (Campbell Scientific) and an LI-7500 CO_2/H_2O gas analyser (Li-Cor). Ground heat flux was measured using a CM3 radiation and energy balance (REBS) ground heat flux plate placed at 0.05 m depth. In the EC technique, the covariance of vertical wind speed with temperature and water vapour is used to estimate the sensible heat (H) and LE fluxes. More information on EC technique can be found in Drexler et al. (2004). Estimates of H and LE were taken at 10 Hz and fluxes were calculated using 30 minute block averages with 2-D coordinate rotation. Since variation of evaporation is commonly perceived as highly dependent on climatic variables, EC measured LE flux is modelled as a function of AT, GT, NR, RH and WS, using both ANNs and GP. The performances of both the data-driven models are also compared with the widely adopted Penman-Monteith (PM) method. The hourly FAO-PM (Temesgen et al., 2005) equation is given by Equation (3.7):

$$ET = \frac{0.408\Delta\left(R_n - G\right) + \gamma \dfrac{37}{AT + 273} WS\left(e^0 - e^a\right)}{\Delta + \gamma(1 + 0.34WS)} \tag{3.7}$$

where R_n is net radiation at the grass surface (MJ m^{-2} h^{-1}), G is soil heat flux density (MJ m^{-2} h^{-1}), Δ is the saturation slope vapour pressure curve at AT (kPa $°C^{-1}$), γ is the psychrometric constant (kPa $°C^{-1}$), e^0 is saturation vapour pressure at air temperature AT (kPa), e^a is the average hourly actual vapour pressure (kPa), and WS is the average hourly wind speed (m s^{-1}).

3.4 Results and Analysis

For both case studies, the functional and terminal set adopted by the GP system in modelling the LE flux are { + ,– , ∗, / } and {NR, GT, AT, RH, WS}, respectively. Prior to modelling the LE flux using GP, the climatic variables were normalized by dividing each variable by their corresponding maximum value. This was done in order to overcome the problem of dimensional inconsistency. These standardized values herein are simply referred to as NR, GT, AT, RH and WS. The performance of the GP model is compared with that of the widely adopted, physically-based PM method and also with the ANN models.

3.4.1 Case Study I

Table 3-2 presents the performance of different models in estimating LE flux for the first case study in terms of RMSE, MARE and R statistics. Both data-driven models (ANNs and GP) performed better than the PM estimates for both the training and testing ranges. During training, the ANN(NR,GT,AT,RH,WS) model performed better than other ANN models in terms of RMSE and *R* statistics. Nevertheless during testing, the ANN(NR,GT) model performed better than other neural networks models in terms of RMSE and MARE (Table 3-2). This indicates that the ANN(NR,GT) model has a better generalization ability (performed better with unseen data) when compared to other ANN models. The optimal equation evolved by the GP system in characterizing the LE flux for Case study I is given by Equation (3.8). Although the terminal set consisted of all five

climatic variables, the GP-evolved equation is a function of NR and GT alone. This demonstrates the ability of GP to identify its own model structure, along with the relevant variables, to characterize the evapotranspiration process. During training, the performance of the GP model was slightly worse than the ANN models. However, during testing, the GP-evolved model performed better than the ANN models in terms of RMSE and MARE (Table 3-2), illustrating the better generalization property of the GP model. In general, the difference in performance statistics between the ANN and GP models is modest (Table 3-2) and, hence, the performance of both techniques is comparable. Figure 3-3 shows the scatter plot between the observed and computed LE by different models.

$$LE = (0.45 - 0.20 * NR) * (0.94 + NR) * (NR + GT - 0.43) \tag{3.8}$$

3.4.2 Case Study II

Table 3-3 presents the performance of different models in estimating EC-measured LE flux for Case study II in terms of RMSE, MARE and R, during both training and testing. Similar to the previous case study, both data-driven models performed better than the PM model in estimating the LE flux (Table 3-3). During training, the model ANN(NR,GT,AT,RH,WS) with all five inputs performed better than the other neural network models in terms of RMSE, MARE and R. However during testing, the ANN(NR,GT,AT,RH,WS) model performed better than the other ANN models in terms of MARE statistics alone (Table 3-3). Nevertheless, the ANN(NR,GT) model performed relatively better than the other ANN models in terms of RMSE and R (Table 3-3). While RMSE gives more weight to high values as it involves the square of

the difference between observed and predicted values, the MARE provides an unbiased error estimate because it gives appropriate weight to all magnitudes of the predicted variable (Karunanithi et al., 1994). Hence, comparing the testing RMSE and MARE statistics of ANN(NR,GT,AT,RH,WS) and ANN(NR,GT) models, the RMSE statistics of both models were similar, while the former model performed better than the latter model in terms of MARE statistics (Table 3-3). This indicates that, while NR and GT alone can explain most of the variance in the LE flux, addition of other climatic variables AT, RH and WS as inputs to the ANN model can help improve the predictive ability of the model for low LE flux values.

The optimal equation found by the GP system reads:

$$LE = 0.475\left(\left\{\left[\left(0.6 \times AT \times WS\right) + 0.78 \times GT\right] \times 0.78 \times GT^2\right\} + NR\right) \tag{3.9}$$

From a terminal set with all five climatic variables, the GP system was robust in evolving the optimal model for characterizing LE as a function of AT, WS, GT and NR alone. Comparing the performance of the GP model with the ANN models (Table 3-3), similar to the previous case study (Table 3-2), during training the performance of the GP model is slightly worse than the ANN models with respect to RMSE (Table 3-3). However, during testing, the GP-evolved model resulted in the smallest MARE and the highest *R* statistics (Table 3-3). The better performance of the GP model during testing signifies its better generalization property. Analysing Equation (3.9), it can be seen that, although AT, WS, GT and NR appear in the GP-evolved model, the equation is

dominated by GT and NR. Figure 3-4 shows the scatter plots between the observed and predicted LE by different models for this case study.

The PM method, which accounts for the influence of vegetation on evapotranspiration, has been used frequently to model the evapotranspiration flux (Abbott et al., 1986). However, the PM method estimates potential reference evaporation, and in reality water is not always freely available (supply limited) to evaporate. Hence, the PM method typically overestimates evapotranspiration (Figure 3-3 (a) and Figure 3-4 (a)) during supply limited conditions and consequently is not directly comparable to their EC-measured actual evapotranspiration counterparts. Nevertheless, the initiative of comparing PM estimates with EC-measured LE flux is to demonstrate the ability of data-driven models in directly modelling the above flux as a function of climatic variables, against the PM method, which would have been used in the absence of such models. The performance of the PM method in the second case study is better than its performance in the first case study. The probable cause of the enhanced model performance is the increased wetness in 2005. Between 1 May and 30 August, rainfall was 227 mm in 2005 compared with 147 mm in 2003. The reduction in water stress as a result of increased precipitation would allow actual evapotranspiration rates to approach potential evapotranspiration estimates using the PM method.

3.5 Discussion

For both case studies, the performance of the GP-evolved model is comparable with the performance of ANN models. However, the problem of identifying the optimal

input combination by the trial-and-error method innate in ANN modelling, can be overcome by the ability of GP in evolving its own model structure with relevant inputs. Also, in the case of the ANN model trained to learn the evapotranspiration process, the knowledge of the process learned by it is represented in the form of a weight matrix, which is difficult to comprehend. However, for the same problem, GP provided an explicit model structure that can help improve our knowledge of the system being modelled. Although many evaporation models use vapour pressure deficit to estimate evaporation, it is of interest to observe that both the GP-evolved models (Equations (3.8) and (3.9)) are not a function of RH. This may be because the land surface states contain the signals of near-surface atmospheric conditions as a result of strong land–atmosphere interaction (Lakshmi and Susskind, 2001; Wang et al., 2004). Hence, the RH would have been a redundant variable for the GP model as it would have learnt the signal of RH embedded in other variables. It can be observed that for both case studies, the GP-evolved models are dominated by NR and GT (Equations (3.8) and (3.9)). This indicates that NR and GT alone can effectively characterize most of the variation in the LE flux. This finding is of particular importance as it considerably reduces the number of climatic variables that need to be measured for modelling EC-measured LE flux.

The rate of evapotranspiration is largely controlled by the available energy and moisture where NR is the driving variable during energy limited conditions and soil moisture is the influential variable during supply limited conditions. Hence, ideally, evapotranspiration should be modelled as a function of climatic variables and soil moisture. However, as indicated earlier, interpolation of soil moisture data to match the

required spatial and temporal resolution of other climatic variables involves large uncertainty. Hence, in this study, evapotranspiration is modelled as a function of readily available climatic variables only. Nevertheless, ground temperature (GT) can be considered as a surrogate variable for soil moisture due to the strong link between soil thermal properties and moisture status. The water content of the top soil layer controls the heat capacity of the soil and, in part, the partitioning of latent and sensible heat. When soils are wet, a slower thermal response is associated with increased evaporation. Conversely, as soils dry, temperature changes are more rapid and latent heat flux declines. The importance of water content in the top layer and its influence on turbulent fluxes and ground temperature has been previously noted (Eltahir, 1998; Wang et al., 2004). It should be noted that the data sets used in deriving the GP-based evapotranspiration equation (Equations (3.8) and (3.9)) are from the spring and summer months only, when the evapotranspiration rate is far greater than during the reminder of the year. Hence the results from this study are of particular importance as they illustrate the major share of annual evapotranspiration.

3.6 Summary and Conclusions

In this study, the utility of genetic programming in modelling the eddy-covariance (EC) measured evapotranspiration flux is investigated. The performance of the GP technique is compared with artificial neural network and Penman-Monteith model estimates. EC measured evapotranspiration fluxes from two distinct case-studies with different topographic conditions were considered for the analysis, and latent heat is modelled as a function of net radiation, ground temperature, air temperature, wind speed

and relative humidity. Results from the study indicate that both data-driven models (ANN and GP) performed better than the Penman-Monteith method. However, the performance of the GP model is comparable with that of ANN models. One of the important advantages of employing GP to model evapotranspiration process is that, unlike the ANN model, GP resulted in an explicit model structure that can be easily comprehended and adopted. From the GP-evolved models, it was found that ground temperature and net radiation dominate the equation for modelling evapotranspiration. This indicates that net radiation and ground temperature alone can represent most of the variation in LE. This finding may help in reducing the number of climatic variables that need to be measured to build parsimonious models and predict LE. In general, it has been found that GP appears to be a promising tool for modelling the evapotranspiration process. The study can be further extended by trying different combinations of mathematical operators in the functional set and by applying the GP to model evapotranspiration from different sites with diverse morphological characteristics.

3.7 Acknowledgements

Funding for this project was provided by the National Science and Engineering Research Council (NSERC) of Canada and the Department of Civil and Geological Engineering, University of Saskatchewan, through the scholarship programme. The genetic programming software, GPLAB, was provided by S. Silva and is available at http://gplab.sourceforge.net.

3.8 References

Abbott, M. B., Bathurst, J. C., Cunge, J. A., O'Connell, P. E. and Rasmussen, J. (1986). "An introduction to the European hydrological system – Système Hydrologique Européen, SHE. 1: History and philosophy of a physically-based, distributed modeling system." *J. Hydrol.,* 87, 45–59.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (ASCE). (2000). "Artificial neural networks in hydrology. II: Hydrologic applications." *J. Hydrol. Eng.,* 5(2), 124–137.

Babovic, V., and Abbott, M. B. (1997). "Evolution of equation from hydraulic data: Part I-Theory." *J. Hydraul. Res.,* 35(3), 1–14.

Babovic, V., and Keijzer, M. (2000). "Genetic programming as model induction engine." *J. Hydroinformatics,* 2, 35–60.

Biftu, G. F., and Gan, T. Y. (2000). "Assessment of evapotranspiration models applied to a watershed of Canadian Prairies with mixed land-uses." *Hydrol. Processes,* 14, 1305–1325.

Blaney, H. F., and Criddle, W. D. (1950). "Determining water requirements in irrigated area from climatological irrigation data." *Soil Conservation Service Technical Paper no. 96.* US Department of Agriculture, Washington DC, USA.

Bowden, G. J., Dandy, G. C., and Maier, H. R. (2005). "Input determination for neural network models in water resources applications. Part 1 – background and methodology." *J. Hydrol.,* 301, 75–92.

Choudhury, B. J., and Monteith, J. L. (1988). "A four-layer model for the heat budget of homogeneous land surfaces." *Quart. J. Roy. Met. Soc.,* 114, 373–398.

Coulibaly, P. (2004). "Downscaling daily extreme temperatures with genetic programming." *Geophys. Res. Lett.,* 31, L16203, doi:10.1029/2004GL020075.

Demuth, H., and Beale, M. (2001). *Neural network toolbox learning. For use with MATLAB*. The Math Works Inc, Natick, Massachusetts, USA.

Drexler, J. Z., Snyder, R. L., Spano, D., and Paw, K. T. (2004). "A review of models and micrometeorological methods used to estimate wetland evapotranspiration." *Hydrol. Processes,* 18, 2071–2101.

Eltahir, E. A. B. (1998). "A soil moisture–rainfall feedback mechanism. 1: Theory and observations." *Water Resour. Res*., 34(4), 765–776.

Flerchinger, G. N., Hanson, C. L., and Wight, J. L. (1996). "Modeling evapotranspiration and surface energy budgets across a watershed." *Water Resour. Res.,* 32(8), 2539–2548.

Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, USA.

French, M. N., Krajewski, W. F., and Cuykendall, R. R. (1992). "Rainfall forecasting in space and time using a neural network." *J. Hydrol.,* 137, 1–31.

Granger, R. J., and Gray, D. M. (1989). "Evaporation from natural non-saturated surface." *J. Hydrol.,* 111, 21–29.

Hameed, T., Mariño, M. A., and Shumway, R. H. (1995). "Evapotranspiration transfer-function-noise modeling." *J. Irrig. Drain. Eng.,* 121(2), 159–169.

Haykin, S. (1999). *Neural networks: A comprehensive foundation*, 2nd Ed.,. MacMillan, New York, USA.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, USA.

Hong, Y. S., and Rosen, M. R. (2002). "Identification of an urban fractured-rock aquifer dynamics using an evolutionary self-organizing modelling." *J. Hydrol.,* 259, 89–104.

Hong, Y. S. T., White, P. A., and Scott, D. M. (2005). "Automatic rainfall recharge model induction by evolutionary computational intelligence." *Water Resour. Res.,* 41, W08422, doi:10.1029/2004WR003577.

Hsu, K. L., Gupta, H. V., and Sorooshian, S. (1995). "Artificial neural network modeling of the rainfall–runoff process." *Water Resour. Res.,* 31(10), 2517–2530.

Jarvis, P. G., and McNaughton, K. G. (1986). "Stomatal control of transpiration: scaling up from leaf to region." *Adv. Ecol. Res.*, 15, 1–49.

Karunanithi, N., Grenney, W. J., Whitley, D., and Bovee, K. (1994). "Neural networks for river flow prediction." *J. Comp. Civ. Eng., ASCE* 8(2), 201–220.

Khu, S. T., Liong, S. Y., Babovic, V., Madsen, H., and Muttil, N. (2001). "Genetic programming and its application in real-time runoff forecasting." *J. Am. Water Resour. Assoc., ASCE*, 8(2), 201–220.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection.* The MIT Press, Cambridge, Massachusetts, USA.

Kumar, M., Raghuwanshi, N. S., Singh, R., Wallender, W. W., and Pruitt, W. O. (2002). "Estimating evapotranspiration using artificial neural network." *J. Irrig. Drain. Eng.,*128(4), 224–233.

Lakshmi, V., and Susskind, J. (2001). "Utilization of satellite data in land-surface hydrology: sensitivity and assimilation." *Hydrol. Processes,* 15(5), 877–892.

MacKay, D. J. C. (1992). "Bayesian interpolation." *Neural Computation* 4(3), 415–447.

Maier, H. R., and Dandy, G. C. (2000). "Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications." *Environ. Modell. Software,* 15(1), 101–124.

Mariño, M. A., Tracy, J. C., and Taghavi, S. A. (1993). "Forecasting reference crop evapotranspiration." *Agric. Water Mgmt.,* 24, 163–187.

Minns, A. W., and Hall, M. J. (1996). "Artificial neural networks as rainfall runoff models." *Hydrol. Sci. J.,* 41(3), 399–417.

Monteith, J. L. (1965). "Evaporation and environment in the state and movement of water in living organisms." In: *Proc. Society of Experimental Biology*, Symposium no. 19, 205–234.Cambridge University Press, Cambridge, UK.

Parasuraman, K., Elshorbagy, A., and Carey, S. (2006). "Spiking-modular neural networks: a neural network modeling approach for hydrological processes." *Water Resour. Res.,* 42, W05412, doi:10.1029/2005WR004317.

Penman, H. L. (1948). "Natural evaporation from open water, bare soil and grass." *Proc. Royal Soc. London* 193, 120–146.

Priestley, C. H. B., and Taylor, R. J. (1972). "On the assessment of surface heat flux and evaporation using large scale parameters." *Mon. Weather Rev.,* 100, 81–92.

Savic, D. A., Walters, G. A., and Davidson, J. W. (1999). "A genetic programming approach to rainfall–runoff modelling." *Water Resour. Mgmt.,* 13, 219–231.

Schwefel, H. P. (1981). *Numerical optimization of computer models*. Wiley, Chichester, UK.

Shamseldin, A. Y. (1997). "Application of neural network technique to rainfall-runoff modeling." *J. Hydrol.,* 199, 272–294.

Shuttleworth, J. W., and Wallace, J. S. (1985). "Evaporation from sparse crops-an energy combination theory." *Quart. J. Roy. Met. Soc.,* 111, 839–855.

Shuttleworth, J. W. (1991). "Evaporation models in hydrology." In: *Land Surface Evaporation* (ed. by T. J. Schmugge & J. André), 93–120. Springer-Verlag: New York, USA.

Silva, S. (2005). GPLAB – a genetic programming toolbox for MATLAB. http://gplab.sourceforge.net

Singh, V. P. (1989). *Hydrologic systems: Watershed modeling*. vol. II. Prentice-Hall, New Jersey, USA.

Stephens, J. C., and Stewart, E. H. (1963). "A comparison of procedures for computing evaporation and evapotranspiration." In: *Publication 62, International Association of Scientific Hydrology*, 123–133. International Union of Geodesy and Geophysics, Berkeley, California, USA.

Souch, C., Wolfe, C. P., and Grimmond, C. S. B. (1996). "Wetland evaporation and energy partitioning: Indiana Dunes National Lakeshore." *J. Hydrol.,* 184, 189–208.

Sudheer, K. P., Gosain, A. K., and Ramasastri, K. P. (2003). "Estimating actual evapotranspiration from limited climatic data using neural computing technique." *J. Irrig. Drain. Eng.,* 129(3), 214–218.

Temesgen, B., Eching, S., Davidoff, B. Z., and Frame, K. (2005). "Comparison of some reference evapotranspiration equations for California." *J. Irrig. Drain. Eng.,* 131(1), 73–84.

Tracy, J. C., Mariño, M. A., and Taghavi, S. A. (1992). "Predicting water demand in agricultural regions using time series forecasts of reference crop evapotrasnpiration." In: *Water Resources Planning and Management: Saving a Threatened Resource-In Search of Solutions* (ed. by M. Karamouz), 50–55. ASCE, New York, USA.

Trajkovic, S., Todorovic, B., and Stankovic, M. (2003). "Forecasting of reference evapotranspiration by artificial neural networks." *J. Irrig. Drain. Eng.,* 129(6), 454–457.

Whigham, P. A., and Crapper, P. F. (2001). "Modelling rainfall–runoff using genetic programming." *Math. Comput. Modell.,* 33, 707–721.

Wang, J., Salvucci, G. D., and Bras, R. L. (2004). "An extremum principle of evaporation." *Water Resour. Res.,* 40, W09303, doi:10.1029/2004WR003087.

Zhang, M., Fulcher, J., and Scofield, R. A. (1997). "Rainfall estimation using artificial neural network group." *Neurocomputing,* 16, 97–115.

Table 3-1 Genetic Programming Parameters

GP parameter	Value
Population size	50
Initialization method	Ramped half-and-half
Sampling method	Roulette
Probability of crossover, P_c	0.6
Probability of mutation, P_m	0.3
Cost function	RMSE
Number of generations	400

Table 3-2 Performance Statistics of Different Models – Case StudyI

Model	Training			Testing		
	RMSE	MARE	R	RMSE	MARE	R
PM	104.6	1.26	0.71	125.3	1.53	0.73
ANN(NR,GT,AT,RH,WS)	50.4	0.50	0.86	69.8	1.02	0.72
ANN(NR,GT,AT,RH)	52.3	0.49	0.85	67.6	0.94	0.71
ANN(NR,GT,AT)	55.0	0.51	0.84	68.5	0.98	0.77
ANN(NR,GT)	56.7	0.53	0.82	66.1	0.93	0.77
GP	57.8	0.54	0.82	65.5	0.92	0.77

Table 3-3 Performance Statistics of Different Models – Case Study II

Model	Training			Testing		
	RMSE	MARE	R	RMSE	MARE	R
PM	83.2	0.50	0.78	53.1	0.38	0.83
ANN(NR,GT,AT,RH,WS)	39.0	0.26	0.87	39.3	0.34	0.84
ANN(NR,GT,AT,RH)	40.1	0.28	0.86	40.0	0.41	0.84
ANN(NR,GT,AT)	39.8	0.28	0.86	42.1	0.38	0.82
ANN(NR,GT)	41.7	0.29	0.85	38.8	0.37	0.85
GP	42.2	0.27	0.84	39.0	0.32	0.86

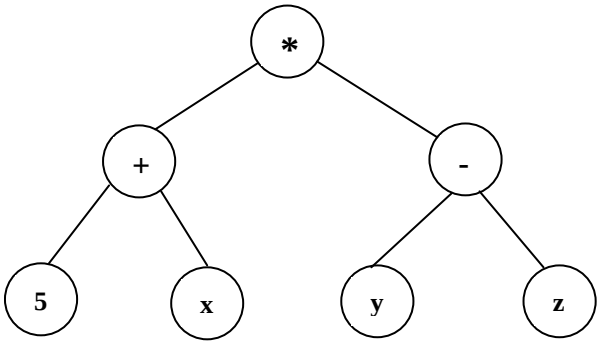

Figure 3-1 Parse Tree Notation

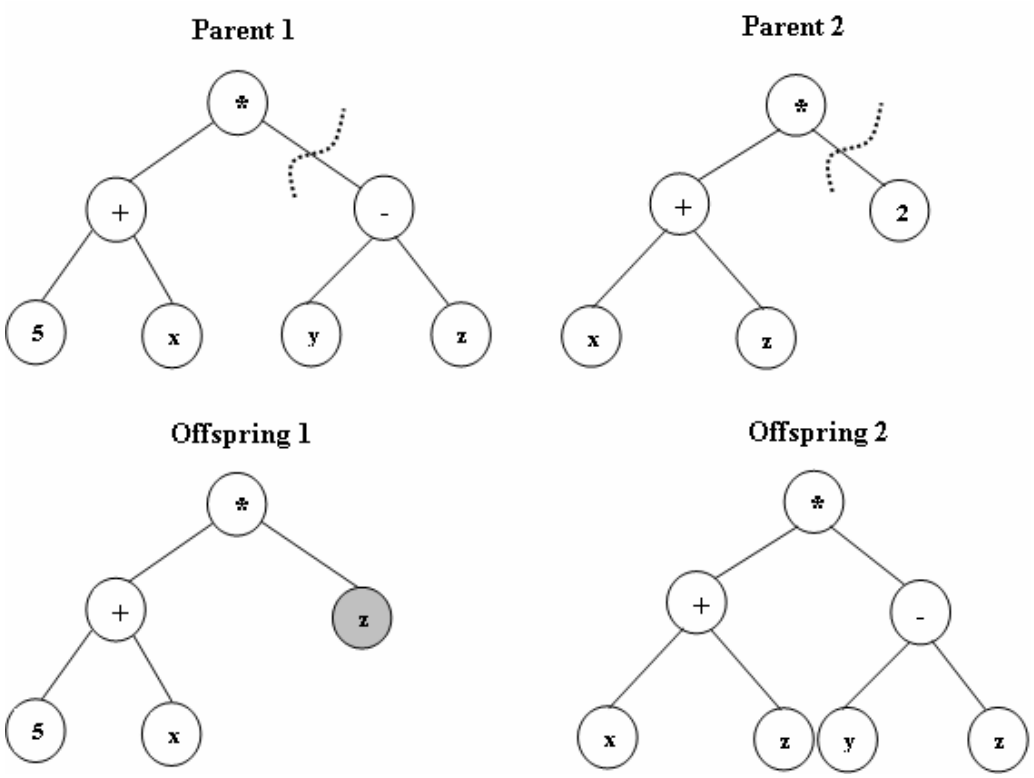

Figure 3-2 Crossover Coupled with Mutation. The dashed line indicates the crossover point and the shaded region represents the mutated node.

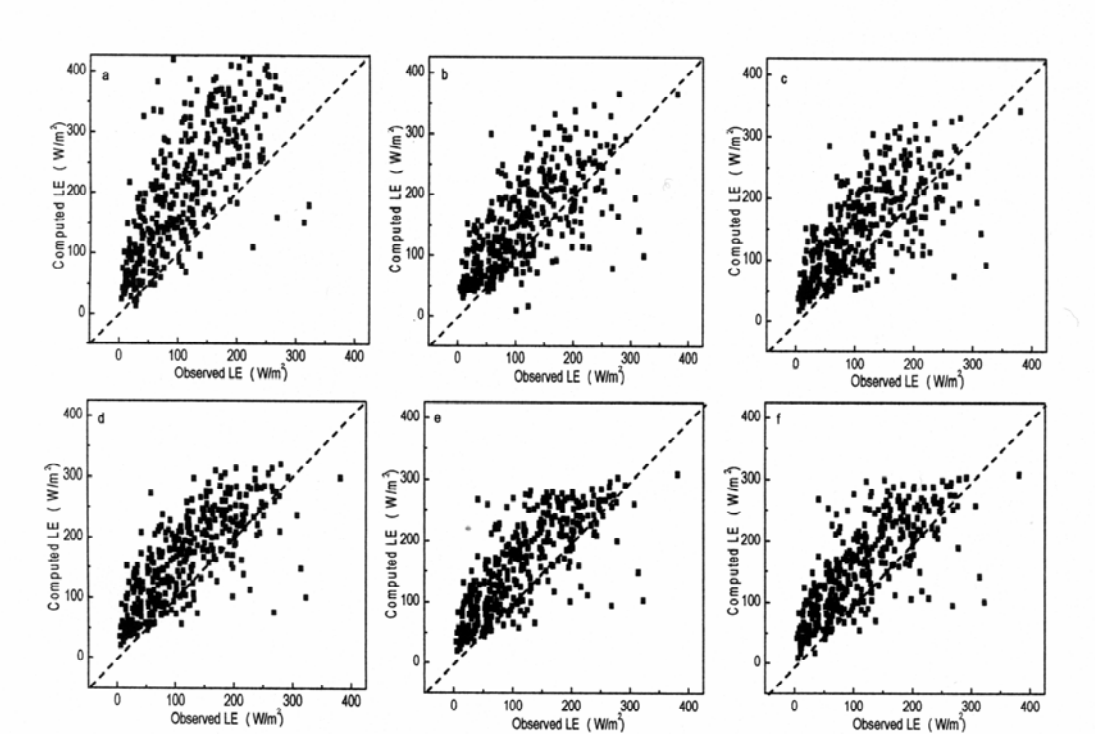

Figure 3-3 Scatter Plots of Observed and Computed LE by (a) PM, (b) ANN(NR,GT,AT,RH,WS), (c) ANN(NR,GT,AT,RH), (d) ANN(NR,GT,AT), (e) ANN(NR,GT), and (f) GP, for Case Study I.

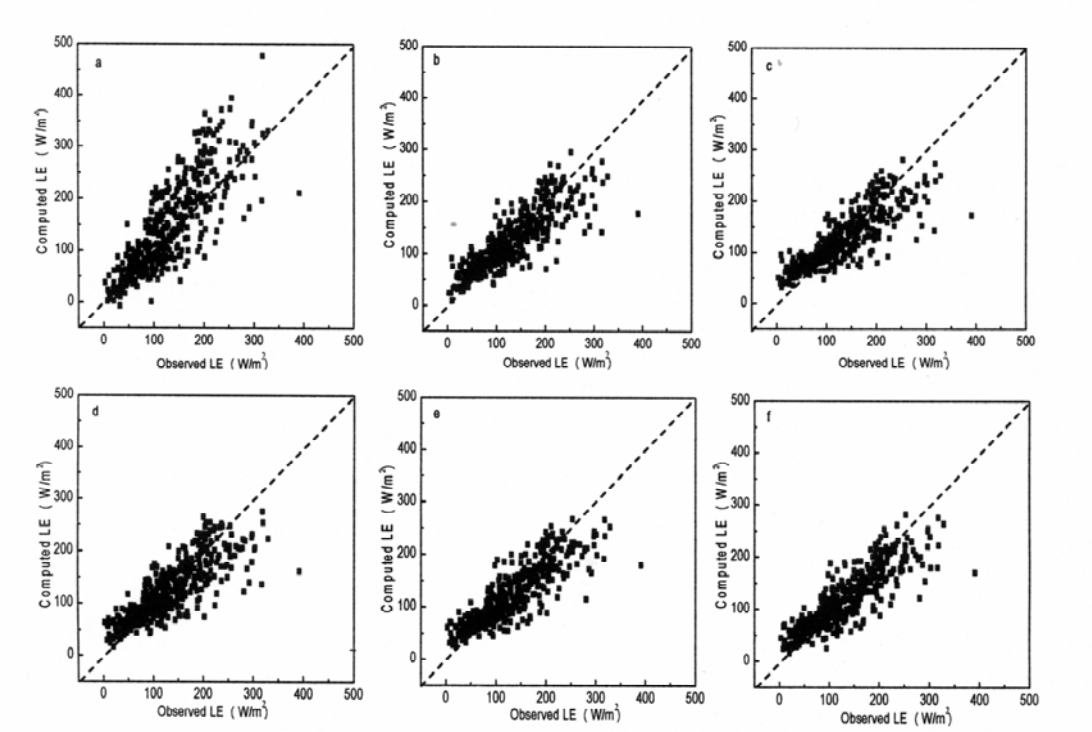

Figure 3-4 Scatter Plots of Observed and Computed LE by (a) PM, (b) ANN(NR,GT,AT,RH,WS), (c) ANN(NR,GT,AT,RH), (d) ANN(NR,GT,AT), (e) ANN(NR,GT), and (f) GP, for Case Study II.

Chapter 4 - Estimating Saturated Hydraulic Conductivity In Spatially-Variable Fields Using Neural Network Ensembles

This chapter has been copyrighted and published as a research paper in the Soil Science Society of America Journal.

Citation: Parasuraman, K., Elshorbagy, A., and Si, B. C. (2006). "Estimating saturated hydraulic conductivity in spatially variable fields using neural network ensembles." *Soil Sci. Soc. Am. J.,* 70, 1851-1859.

Contribution of the PhD candidate

Model conceptualization was instigated by Kamban Parasuraman, Dr. Amin Elshorbagy, and Dr. Bing Cheng Si. Kamban Parasuraman carried out computer program development and simulation, with Dr. Amin Elshorbagy and Dr. Bing Cheng Si providing guidance on various aspects of the study. The dataset used in this study was provided by Dr. Bing Cheng Si. The text of the published paper was created by Kamban Parasuraman, with Dr. Amin Elshorbagy and Dr. Bing Cheng Si critically reviewing the manuscript.

Contribution of this chapter to the overall study

The previous two chapters' highlight some of the possible methods and tools that can be adopted to promote transparency in the way data-driven models arrive at a decision in modeling the hydrological processes. Nevertheless, this chapter and the subsequent one identify ways for improving the reliability of the data-driven models. In

this chapter, the improvement in the reliability that can be achieved by adopting a local-scale model, as against a global-scale model, is evaluated by developing pedotransfer functions (PTFs) to characterize the saturated hydraulic conductivity of soils. Local-scale neural network-based pedotransfer functions were developed and compared with a published global neural network model, ROSETTA. The local-scale models are shown to be more reliable than the global-scale models. Also, an algorithm for reducing both the bias and variance of the neural networks-based PTFs is identified in this study.

4.1 Abstract

Modeling contaminant and water flow through soil requires accurate estimates of soil hydraulic properties in field scale. Although artificial neural networks (ANNs) based pedotransfer functions (PTFs) have been successfully adopted in modeling soil hydraulic properties at larger scales (national, continental, and intercontinental), the utility of ANNs in modeling saturated hydraulic conductivity (K_s) at a smaller (field) scale has rarely been reported. Hence, the objectives of this study are (i) to investigate the applicability of neural networks in estimating K_s at field scales, (ii) to compare the performance of the field-scale PTFs with the published neural networks program *Rosetta,* and (iii) to compare the performance of two different ensemble methods, namely Bagging and Boosting in estimating K_s. Datasets from two distinct sites are considered in the study. The performances of the models were evaluated when only sand, silt, and clay content (SSC) were used as inputs, and when SSC and bulk density ρ_b (SSC+ ρ_b) were used as inputs. For both datasets, the field scale models performed better than *Rosetta*. The comparison of field-scale ANN models employing bagging and boosting algorithms indicates that the neural network model employing the boosting algorithm results in better generalization by reducing both the bias and variance of the neural network models.

4.2 Introduction

Estimation of the hydraulic properties of soils is of paramount importance for modeling contaminant and water flow through the vadose zone. Hydraulic properties also

play an important role in partitioning the rainfall into runoff and soil moisture components. Soil hydraulic properties are usually measured in a laboratory using representative soil samples from the study area. Since the hydraulic properties exhibit large variations within a spatial domain, large numbers of soil samples are required to characterize the hydraulic properties of the study area. Laboratory estimates of hydraulic properties are complex and time consuming; therefore, the interest in using PTFs to estimate the hydraulic property of the soil is increasing (Rawls and Brakensiek, 1983; Cosby et al., 1984; Saxton et al., 1986; Vereecken et al., 1990; van Genuchten et al., 1992; Leij et al., 2002).

Pedotransfer functions relate hydraulic properties to easily measurable or more widely available soil parameters (Bouma, 1989). A detailed review of different pedotransfer functions is given by Wösten et al. (2001). PTFs models include traditional regression models (Wösten et al., 1995; Rawls et al., 1991) and ANNs (Schaap et al., 1998; Schaap and Bouten, 1996; Pachepsky et al., 1996; Minasny et al., 1999). A detailed review of ANNs and their application in predicting soil hydraulic properties can be found in Tamarai and Wösten (1999).

Schaap et al. (1998) showed that ANNs performed better than four published pedotransfer functions in estimating water retention data and six published pedotransfer functions in estimating the saturated hydraulic conductivity (K_s). The dataset used by Schaap et al. (1998) is derived from 4515 laboratory samples taken from 30 sources in the USA. Pachepsky et al. (1996) showed that the neural networks and regression models

performed similarly in predicting the water retention parameters based on a dataset of 230 soil samples. Minasny and McBratney (2002) proposed a new objective function for neural network training, which predicted the parameters of the parametric model and optimized the PTF to match the observed and measured water contents. The study made use of 862 soil samples collected across Australia. Minasny and McBratney (2002) showed that their new objective function improved the performance of the neural network model when compared to the models employing traditional objective functions, in which the networks were optimized to fit the model parameters. Schaap et al. (2001) proposed a computer program, *Rosetta*, which implemented five hierarchical pedotransfer functions for the estimation of water retention and the saturated and unsaturated hydraulic conductivity. *Rosetta* is based on neural network analyses combined with the bootstrap method. The dataset used for constructing *Rosetta* was derived from soils in temperate to subtropical climates of North America and Europe. Most of the above discussed studies include a large number of samples obtained from a national scale, and it has been demonstrated that the ANNs are robust in predicting the hydraulic properties.

Traditionally, hydraulic properties are estimated from PTFs that were developed elsewhere (Tietje and Hennings, 1996; Tietje and Tapkenhinrichs, 1993). Hence, PTFs have been developed at various scales, including national (Nemes et al., 2003), continental (Wösten et al., 1999; Nemes et al., 2003), and intercontinental scales (Nemes et al., 2003). Nemes et al. (2003) showed that the PTFs developed at one scale were not suited for other scales. Moreover, they suggested that deriving PTFs from a small set of relevant data, when available, was more appropriate than using PTFs derived from a large

but more general dataset. Also, according to Bastet et al. (1999), a particular PTF cannot be applied to the entire soil horizon; therefore, researchers should establish the validity of a PTF before adopting it. PTFs developed at a large scale are best suited for global climate modeling, but might be of little use for modeling chemical transport and soil water balance on a farm field. The importance of the above considerations can be seen in the following example. Romano and Palladino (2002) examined the prediction of soil hydraulic properties along two linear transects based on soil physical properties and terrain information. Although efforts have been made to develop PTFs at large scales, little research has been conducted to evaluate the performance of ANNs in estimating saturated hydraulic conductivity (K_s) at field scale. Moreover, although the utility of the bagging algorithm in improving the generalization ability of ANNs models has been reported in various studies (Schaap et al., 1998; Schaap et al., 2001; Nemes et al., 2003; Minasny et al., 2004), the ability of more versatile boosting algorithms (Schapire, 1990; Freund and Schapire, 1996) in improving the generalization ability of ANNs models in predicting K_s has not been investigated. Compared to bagging algorithm, boosting algorithm improves performance by producing a series of neural networks trained with a different distribution of the original training data.

The general objective of this study is to investigate the applicability of ANN-based pedotransfer functions at a field scale. The specific objectives include (1) determining the best combination of inputs in predicting K_s at field-scale, (2) comparing the performance of the field-scale PTF with the published neural networks program

Rosetta, and (3) evaluating the relative performance of field-scale ANNs models employing bagging and boosting algorithms.

4.3 Materials and Methods

4.3.1 Artificial Neural Networks

Feed-forward neural networks (FF-NNs) are the most widely adopted network architecture for the prediction and forecasting of geophysical variables (Maier and Dandy, 2000). Typically, FF-NNs consist of three layers: the input layer, hidden layer, and output layer. The number of nodes in the input layer corresponds to the number of inputs considered for modeling the output. The input layer is connected to the hidden layer with weights that determine the strength of the connections. The number of nodes in the hidden layer indicates the complexity of the problem being modeled. The hidden layer nodes consist of the activation function, which helps in nonlinearly transforming the inputs into an alternative space where the training samples are linearly separable (Brown and Harris, 1994). The most commonly used activation function is the sigmoidal transfer function as it is a bounded, monotonic, nondecreasing function that provides a graded, nonlinear response. The hidden layer is connected to the output layer. Detailed review of ANNs and their application in water sciences can be found in Maier and Dandy (2000) and in ASCE Task Committee on Application of Artificial Neural Networks in Hydrology (2000 a, b).

The structure of the three-layered FF-NN used in this study is shown in Figure 4-1. The neural network model consists of *'j'* input neurons, *'k'* hidden neurons, and *'l'* output neurons. Symbolically, the ANN architecture shown in Figure 4-1 can be represented as *ANN(j,k,l)*. The FF-NN adopted in this study makes use of the log-sigmoidal activation function in both the hidden layer and the output layer. In Figure 4-1, W_{kj} represents the connection weight between the j^{th} input neuron and k^{th} hidden neuron. Similarly, W_{lk} represents the connection weight between the k^{th} hidden neuron and l^{th} output neuron. Parameters b_k and b_l represent the bias of the corresponding hidden and output layer neurons. The role of bias in a neuron is to displace the original functional domain by a magnitude equal to that of the bias and thereby translate the area of influence to its activation state. If x_j represents the input variables and y_l represents the output variables, then the inputs are transformed to output by the following equations (Haykin, 1999):

$$y_l = f_1\left[\sum_{k=1}^{K} w_{lk} f_1\left(\sum_{j=1}^{J} w_{kj} x_j + b_k\right) + b_l\right] \tag{4.1}$$

$$f_1(p) = \frac{1}{\left(1 + e^{-p}\right)} \tag{4.2}$$

where $f_1(.)$ represents the log-sigmoidal activation function. The log-sigmoidal activation function helps in squashing the inputs between 0 and 1. One of the important issues in the development of a neural network model is the determination of optimal number of hidden neurons that can satisfactorily capture the nonlinear relationship existing between the input and the output variables. The number of neurons in the hidden

layer is usually determined by the trial-and-error method with the objective of minimizing the cost function (ASCE Task Committee on Application of Artificial Neural Networks in Hydrology, 2000a). The typical cost function used in training FF-NNs involves minimizing the mean sum of squares of the network errors (MSE). In Equation (4.3), y_i and $y_i^{'}$ represent the measured and computed counterparts, and n represents the number of training instances. A systematic search of different network configurations and user-adjustable parameters is carried out to ascertain the optimal network architecture, with the objective of minimizing the cost function. The optimal network architecture is the one which results in the least cost function.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^{'}\right)^2 \tag{4.3}$$

The development of a neural network model demands two operations, namely, (i) training and (ii) testing. Training is a process by which the connection weights between different layers and the bias values of the neural networks are optimized by minimizing the cost function. Since *Rosetta* uses Levenberg-Marquardt algorithm (Demuth and Beale, 2001), for a rational comparison of the proposed field-scale models with *Rosetta*, the same algorithm is adopted in this study to determine the optimal combination of connection weights and biases of the field-scale models. Once trained, the neural network model can be tested on an independent dataset that has not been used during the training process. More information on the Levenberg-Marquardt algorithm can be found elsewhere (Haykin, 1999).

One of the important properties of any neural network model is its generalization ability; i.e., the ability of the neural network model to accurately predict the data that are not used for training the model. Recent theoretical and empirical studies have shown that the generalization ability of the neural network model can be improved by combining several neural network models in redundant ensembles. Hence in this study, the ANNs model is coupled with "bagging" (Breiman, 1996) and "boosting" (Schapire, 1990) algorithms, where several redundant ensembles of ANNs, created based on a statistical resampling technique (Efron and Tibshirani, 1993), are combined together to generate a unique output.

4.3.2 Bagging

Bagging (Breiman, 1996) is an acronym for "bootstrap aggregation." Using bagging, various datasets are generated from multiple realizations of the training dataset and these datasets are trained using different neural network models. The outputs from each of the neural network models are combined together to give a unique output. Moreover, bootstrapping allows the generation of an uncertainty estimate for each predicted value, which in turn aids the evaluation of the reliability of the model. The following paragraph outlines the methodology for carrying out bagging.

Suppose the training dataset T consists of N instances $(x_1, y_1),\ldots,(x_N, y_N)$, where x and y are input and output variables respectively. It is desired to obtain B bootstrap datasets. As a first step, each instance in T is assigned a probability of $1/N$, and the training set for each of the bootstrap member T_B is generated by sampling with

replacement N times from the original dataset T using the above probabilities. Hence each bootstrap dataset T_B may have many instances in T repeated several times, while other instances may be left out. Individual neural network models are then trained on each of T_B. Therefore for any given input vector, the bootstrap algorithm provides B different outputs. The bagging estimate is then calculated by finding the mean of B different model predictions and the bagging uncertainty is estimated by finding the standard deviation of the B different model predictions.

4.3.3 Boosting

Compared to bagging, boosting algorithms (Schapire, 1990; Freund and Schapire, 1996) achieve improved performance by producing a series of neural networks trained with a different distribution of the original training data. The algorithm trains the initial neural networks with the original dataset and the training datasets for successive neural network models are assembled based on the performance of the current neural network model. If predicted values obtained from the current neural network model differ significantly from their observed values, the observed values will have higher probability of being selected in successive neural network models. In this way, the network is focused on learning hard patterns, thereby improving the performance of the neural network model. In this study, the boosting algorithm ADABoost.R2 proposed by Drucker (1999) is adopted. ADABoost.R2 is a variation of the adaptive boosting algorithm, ADABoost.R proposed by Freund and Schapire (1996). Drucker (1999) showed that, in most cases, the ADABoost.R2 algorithm performed better than bagging in terms of

prediction error when applied to ANNs. The ADABoost.R2 algorithm (Drucker, 1999) is detailed below.

Assume that the training dataset T consists of N instances $(x_1, y_1),\ldots,(x_N, y_N)$, where x and y are input and output variables respectively. Initially each value in the dataset is assigned the same probability value so that each instance in the initial dataset has an equal chance of being sampled in the first training set; i.e., sampling distribution, $D_t(i)$ at step $t=1$, is equal to $1/N$, over all i, where $i=1$ *to* N. Iterate the following, while the average loss $\bar{L}$, defined below, is less than 0.5 or a preset number of networks *(t)* are constructed.

1. Populate the new training set $NewT_t$ from the original training dataset T using the distribution D_t.
2. Construct a new network k_t, and train it using $NewT_t$.
3. Calculate the maximum loss, L_{max}, between the actual value and the network output $k_t(x_i, y)$, over the initial training set T where:

$$L_{\max} = \sup\left(\left|k_t(x_i, y) - y_i\right|\right), \text{ over all i} \tag{4.4}$$

   Where sup() represents the maximum value of a set.

4. Calculate the individual L_i, loss for each element in the training set:

$$L_i = 1 - \exp\left[-\frac{\left|k_t(x_i, y) - y_i\right|}{L_{\max}}\right] \tag{4.5}$$

5. Calculate the weighted average loss, $\bar{L}$:

$$\bar{L} = \sum_{i=1}^{N} L_i D_t(i) \tag{4.6}$$

6. Set β_t

$$\beta_t = \frac{\bar{L}}{1-\bar{L}} \tag{4.7}$$

7. Update the distribution D_t:

$$D_{t+1}(i) = \frac{D_t(i)\beta_t^{(1-L_i)}}{Z_t} \tag{4.8}$$

Where Z_t is a normalization factor chosen such that D_{t+1} is a distribution.

8. Increment t by 1

For any given input vector, the boosting algorithm provides B different outputs, similar to bagging. Hence the boosting estimate and boosting uncertainty are estimated by finding the mean and standard deviation of the B different model predictions. The main difference between the neural network models employing the bagging and boosting algorithm is as follows: in the boosting algorithm, the distribution of the training set changes adaptively based on the performance of the previously created network, while the bagging algorithm changes the distribution of the training set stochastically. Although the boosting algorithms have better generalization ability than the bagging algorithms, the latter algorithm has the advantage of training the ensembles independently, hence in parallel. In this study, in-house codes for bagging and boosting algorithms were developed using MatLab (the Mathworks, Lowell, MA). The performances of the models were evaluated when only sand, silt, and clay contents (SSC) were used as inputs, and when SSC and bulk density ρ_b (SSC+ ρ_b) were used as inputs. Using bagging and

boosting algorithms, several redundant ensembles of ANNs, were created based on a statistical resampling of inputs (SSC and SSC+ ρ_b), and are combined together to generate a unique output (K_s). For both bagging and boosting algorithms, the optimal ensemble size, *B*, was found to be 30, using the trial-and-error method. Also, the number of hidden neurons in the networks employing bagging and boosting algorithms was determined using the trial-and-error method. Results from trial-and-error analysis indicated that the predictability of neural networks did not improve significantly with use of more than two hidden neurons. Hence the optimal number of hidden neurons for neural network models employing both bagging and boosting algorithms was two. Herein, the neural network model using the bagging algorithm will be referred to as *Field(Bagging)* and that using boosting algorithm will be referred to as *Field(Boosting)*.

4.3.4 Performance Evaluation

The performances of the different models are evaluated based on (i) root mean square error (RMSE), (ii) mean absolute relative error (MRE), and (iii) mean residual (MR). RMSE, MRE, and MR statistics are calculated using Equations. (4.9), (4.10), and (4.11) respectively, where *n* represents the number of instances presented to the model and y_i and $y_i^{'}$ represent measured and computed K_s respectively.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^{'})^2} \tag{4.9}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - y_i^{'}}{y_i}\right| \tag{4.10}$$

$$MR = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^{'}\right) \tag{4.11}$$

Since logarithmic values of K_s are considered, the corresponding RMSE, MRE, and MR statistics are dimensionless. Each of the above performance statistics provides different information about the predictive ability of the models. The RMSE statistic indicates only the model's ability to predict away from the mean (Hsu et al., 1995). RMSE gives more weight to high K_s values because it involves square of the difference between observed and predicted values. The MRE provides an unbiased error estimate because it gives appropriate weight to all magnitudes of the predicted variable. The closer to one is the ratio of predicted to measured, the smaller the MRE. This aspect of relative error is found to give a more appropriate assessment and comparison of different models (Legates and McCabe, 1999). The MR is a measure of prediction bias, with a negative and positive value of MR indicating underprediction and overprediction, respectively.

The best model should be unbiased (MR=0), have the smallest MRE and have smallest overall dispersion (RMSE). In addition, the uncertainty or standard deviation among realizations of predicted values using Bagging or Boosting should be small. The uncertainty estimate, unlike the RMSE, MRE, and MR statistics, indicates the reliability of the K_s estimates. When no independent hydraulic data are available, calculation of RMSE, MRE, and MR statistics is not possible because it requires the measured K_s value. However, uncertainty estimates can still provide the measure of reliability for the K_s, predicted by model. In this study, we consider these criteria equally important. Hence, in order to access the overall performance of each model, a rank score technique (Pandey

and Nguyen, 1999; Shu and Burn, 2004) is adopted. To calculate the rank score, the models are ranked from best to worst according to the performance indices. Supposing that there are p models under consideration, a score of 1 is assigned to the best model and p for the worst model. For each model, the scores for the different performance indices are summed to obtain the overall performance score R_o for the model. Supposing that there are q indices, then the overall rank scores are in the range [q, pq]. R_o is then normalized to obtain the normalized rank score R_n, using the following equation, where the normalized rank scores are in the range [0, 1], and an R_n close to 1 represents a model with good performance.

$$R_n = \frac{pq - R_o}{pq - q} \tag{4.12}$$

4.3.5 Site Description and Sampling

4.3.5.1 Case study I: Smeaton

The Smeaton research site is located at Smeaton, SK, Canada (53° 40' N and 104° 58' W). The soil at the site is classified as Gleyic Luvisol with texture dominated by sandy loam developed from glacio-fluvial and fluvial-lacustrine sands and gravels. The topography of the site is gently undulating and the climate is classified as cold and sub-humid. The long-term annual temperature, rainfall, and potential evapotranspiration are 0.1 °C, 393 mm, and 530 mm, respectively (Anderson and Ellis, 1976).

A north-south transect of 384-m length was established on a gently sloping land with a variable texture and organic carbon content (Si and Zeleke, 2005; Zeleke and Si, 2005). After preliminary observations, a 3-m sampling interval was marked along the transect and core samples were collected in September, 2003 using 54-mm-diameter by 60-mm-long aluminium rings. All the 128 cores were used to determine the sand, silt, and clay content; and the bulk density (ρ_b) of the soil. Hydrometer method (Gee and Bauder, 1986) was used to determine the particle size distribution. Saturated hydraulic conductivity (K_s) of the undisturbed core samples was determined using the constant head method (Klute and Dirksen, 1986). Of the 128 samples, two were considered outliers and the remaining dataset (126 samples) is split into training and testing datasets.

The dataset is split in such a way that every third instance appears in the testing set and the remaining instances make up the training set. This data segregation is carried out in order to account for the spatial variability of the soil properties within both the training and testing datasets. The statistics of the entire dataset, along with the dataset used for training and testing are given in Table 4-1. Prior to modeling K_s using neural networks, the values of K_s are logarithmically transformed to avoid bias towards high conductivities. While sand content and ρb showed the least variation, clay content showed the highest variation. Moreover the training and testing datasets have similar statistical properties (Table 4-1).

4.3.5.2 Case study II: Alvena

The Alvena research site is located at Alvena, SK, Canada (52° 31' N and 106° 01' W). The dominant soil type is an Aridic Ustoll and the landscape is classified as hummocky. The long-term annual temperature, rainfall, and potential evapotranspiration are 2.2 °C, 350 mm, and 624 mm, respectively (Si and Farrell, 2004). Undisturbed soil samples are collected along transect of 612 m length with a variable texture. The sand, silt, and clay content, along with the bulk density, were determined from these soil samples. Similar to the previous case study, particle size distribution was determined based on the hydrometer method, and K_s was determined using the constant head method. Of the 78 samples, the training and testing datasets were selected similarly to the previous case-study. The training and testing sets consists of 52 and 26 samples respectively.

The statistics of the entire dataset, along with the dataset used for training and testing, are given in Table 4-2. While silt content ranged from 46 to 63%, clay content ranged from 20 to 41%. Compared to the previous case study, the $\log_{10}(K_s)$ for the Alvena site showed higher variability. The coefficient of variation (CV) of $\log_{10}(K_s)$ for Smeaton and Alvena dataset is 0.14 $\log_{10}$(cm d^{-1}) and 0.24 $\log_{10}$(cm d^{-1}), respectively. The statistics of training and testing dataset are similar (Table 4-2).

4.4 Results and Discussion

For the Smeaton case-study, the performance statistics of different models, when only three (SSC) inputs were used and when four (SSC+ ρ_b) inputs were used in

predicting K_s, are presented in Table 4-3. Since Wösten (1990) recommended that the use of indirect methods for estimating hydraulic properties should be accompanied by the uncertainty of the estimations, the average uncertainty of the predicted K_s during both training and testing is also reported in Table 4-3. Along with RMSE, MRE, and MR statistics, the uncertainty statistics are also considered in calculating the rank score. The rank score presented in Table 4-3 is evaluated based on the performance of different models during testing.

When SSC was used as inputs for training, the field-scale models performed better than *Rosetta* (Table 4-3). This is expected because *Rosetta* is trained outside the field-scale dataset. However, the field-scale models also outperformed *Rosetta* in the testing set. The RMSE, MRE, and MR statistics achieved by *Rosetta* model were 0.26, 0.11, and 0.15 respectively, with an uncertainty value of 0.16 $\log_{10}$(cm d^{-1}). The field-scale models resulted in relatively smaller MR values than that of the *Rosetta*, indicating that the field-scale models were less biased. Also, the field-scale models resulted in smaller uncertainty values, thereby imparting more confidence to the predicted values. The field-scale models using different algorithms, the Field(Bagging) and Field(Boosting), resulted in similar MREs. Nevertheless the Field(Boosting) model performed relatively better in terms of RMSE and MR. This is consistent with the findings of Drucker (1999), who reported that the ADABoost.R2 algorithm had better generalization property than the bagging algorithm. The Field(Boosting) algorithm resulted in an RMSE, MRE, and MR of 0.22, 0.09, and 0.07 respectively. Also, a least

uncertainty value of 0.04 $\log_{10}$(cm d^{-1}) was achieved by both Field(Boosting) and Field(Bagging) models.

Along with SSC, when ρ_b was also used as one of the inputs (SSC+ ρ_b) in predicting K_s, in general the performance of all the models improved considerably in terms of RMSE, MRE, and MR. With RMSE, MRE, and MR statistics of 0.19, 0.08, and -0.07, *Rosetta* showed the maximum improvement when ρ_b was added as input (Table 4-3). The *Rosetta* model, which overpredicted (positive MR) K_s when SSC was used as inputs, resulted in underprediction (negative MR) when SSC+ ρ_b were used as inputs. Also the uncertainty in *Rosetta* model estimates dropped from 0.16 $\log_{10}$(cm d^{-1}) to 0.12 $\log_{10}$(cm d^{-1}) when ρ_b was added. However, opposite results were obtained in the case of field-scale models. When ρ_b was added to the inputs, the estimated uncertainty increased from 0.04 $\log_{10}$(cm d^{-1}) to 0.11 $\log_{10}$(cm d^{-1}) in the case of Field(Bagging) and from 0.04 $\log_{10}$(cm d^{-1}) to 0.10 $\log_{10}$(cm d^{-1}) in the case of Field(Boosting). Comparing the performance of *Rosetta* with the field-scale models, *Rosetta* resulted in the least RMSE statistics. However, the uncertainty estimates are larger than the field-scale models. The Field(Boosting) model resulted in the least MR and uncertainty statistics of 0.04 and 0.10 $\log_{10}$(cm d^{-1}). In general the overall performance of different models, as measured by their rank scores, indicated that the Field(Boosting) model performed better than other models, when either SSC or SSC+ρ_b was used as inputs. Figure 4-2 and Figure 4-3 show the scatter plots between the measured and computed K_s using different models when SSC and SSC+ ρ_b were used as inputs. In general, Rosetta under-predicted K_s when SSC were used as inputs and over-predicted K_s when SSC+ ρ_b were used as inputs. However,

the prediction trends of both the field-scale models were similar for both the input conditions (Figure 4-2 and Figure 4-3).

For the Alvena case-study the performances of different models as measured by RMSE, MRE, MR, along with uncertainty estimates, when SSC or SSC+ ρ_b are used as inputs, are presented in Table 4-4. As with the previous case study, the rank score calculated based on the performance of different models during testing is also presented in Table 4-4. When SSC alone was used as inputs, the field-scale models outperformed *Rosetta* during both training and testing (Table 4-4). MR statistics indicated that the field-scale models were less biased than *Rosetta*. *Rosetta* overpredicted K_s, and the field-scale models underpredicted K_s. The uncertainty estimate was 0.10 $\log_{10}$(cm d^{-1}) for *Rosetta*, 0.09 $\log_{10}$(cm d^{-1}) for Field(Bagging) and 0.10 $\log_{10}$(cm d^{-1}) for Field(Boosting). The Field(Boosting) model performed better than the Field(Bagging) model in terms of MR. However, the Field(Bagging) model resulted in smaller uncertainty value than Field(Boosting) model. Based on the rank score, the performances of both the field-scale models are similar.

When SSC+ ρ_b were used in estimating K_s, the field-scale models again outperformed *Rosetta* (Table 4-4). *Rosetta* resulted in RMSE, MRE, and MR estimates of 0.86, 0.42, and 0.60 respectively. An uncertainty of 0.12 $\log_{10}$(cm d^{-1}) was achieved by the *Rosetta* model. The Field(Boosting) model performed better than the Field(Bagging) model and the Field(Boosting) model resulted in the maximum rank score (Table 4-4). This illustrates the superior performance of the Field(Boosting) model in predicting K_s.

Moreover it should be noted that the addition of ρ_b as one of the inputs resulted in deterioration of the field-scale models performance during testing, although there was significant improvement during training (Table 4-4). This illustrates that the generalization property of the field-scale models is affected when ρ_b is considered as one of the inputs. Nevertheless, the addition of ρ_b as one of the inputs to the *Rosetta* model improved its performance during testing in terms of RMSE and MR, but deteriorated its performance in terms of MRE and the uncertainty estimate. The reason for the poor performances in the field-scale and *Rosetta* models is that ρ_b is poorly correlated to K_s at the Alvena site (R^2=0.01). Figure 4-4 and Figure 4-5 show the scatter plots between the measured and computed K_s using different models when SSC and SSC+ ρ_b were used as inputs. From Figure 4-4 and Figure 4-5, it can be seen that the Rosetta model performed poorly in predicting K_s. Nevertheless, the performance of the local models was relatively better.

In general, it is observed that the performance of the neural network models in estimating K_s for the Smeaton dataset was better (less prediction errors) than that of the Alvena dataset. This is because soil hydraulic property depends on soil texture and soil structure. Soil structure in a sandy soil is dominantly single grained (or sometimes referred to as structureless). Hence substantial difference in K_s, due to soil structure, is unlikely in sandy soils. Nevertheless, soil structure in clay loam soil can be blocky, which in addition to soil texture, can introduce substantial difference in K_s. Since the neural network models consider only the soil texture, the better performance of the neural network models in sandy soils is expected.

Although the field-scale models outperformed *Rosetta* in both cases of the Smeaton (Table 4-3) and the Alvena (Table 4-4), the better performance of the field-scale models against *Rosetta* was more pronounced in the case of Alvena than in Smeaton. At the Smeaton site, the field-scale models had around 11 % reduction in RMSE, 20% reduction in MRE, and 47% reduction in MR and 75% reduction in uncertainty when SSC were used as input. At the Alvena site, field-scale models had about 45% reduction in RMSE, 26% reduction in MRE, 74% reduction in MR, and 10% reduction in uncertainty when SSC were used as inputs. This may be attributed to the following reason. The soil type is sandy loam in Smeaton and silty-clay-loam in Alvena. Compared to the silty clay-loam soils, sandy soils are better represented in the training dataset of *Rosetta* (Schaap et al., 2001). Hence the performance of *Rosetta* was relatively better in the Smeaton case-study. The above finding is of particular significance because it reiterates the importance of the choice of proper training dataset. It can be concluded that the neural network model trained even on a small set of relevant data, when available, is better than training the neural network model with large but more general dataset. This finding is supported by Nemes et al. (2003). Moreover, the field-scale models are more parsimonious than *Rosetta* as the number of hidden neurons is six in *Rosetta*, and is two in the field-scale models. For both the case studies, the inclusion of ρ_b as one of the input parameters to the field-scale models, improved the performance of the models in terms of error estimates. However the uncertainty of the model predictions increased. Hence for the field-scale models, SSC is found to be the optimal combination of inputs.

As hypothesized, the boosting algorithm performs better than the bagging algorithm, which has been conventionally adopted in neural network modeling of soil hydraulic parameters. For both the case studies, the Field(Boosting) model resulted in a considerably less MR value than the Field(Bagging) model. This illustrates that the neural network model using boosting algorithm is less biased. This can be attributed to the ability of the boosting algorithm to focus and learn hard patterns, which in turn improves the performance of the neural network models. Unlike the bagging algorithm, which is largely a variance reduction method, the boosting algorithm is shown to reduce both bias and variance of the model. After each network in the ensemble is trained, the training samples with large errors have their weights increased while the training samples with small errors have their weights reduced for the purpose of training the next network in the ensemble. In this way, the boosting algorithm attempts to reduce the bias of the most recent network in the ensemble by focusing more on the training samples that have larger prediction errors.

In this study, we evaluated the performance of *Rosetta*, Field(Bagging), and Field(Boosting) models based on their ability in predicting the saturated hydraulic conductivity at field scales. More uncertainty analysis regarding the applicability of the neural networks predicted values in modeling the hydrological processes is beyond the scope of this study. Global models have wide applicability, but are found to perform poorly in field scale studies. Nevertheless, the most practical environmental and agricultural applications are at field scales. In this regard, the study is unique in that the performance of the ANNs models in predicting K_s at field scale is explored. While one

cannot extrapolate field-scale models to drastically different fields, field-scale models reduce the number of measurements required for that field. Also, in this study we illustrated the robustness of boosting algorithms in improving the generalization property of neural network models. Compared to the networks implementing the bagging algorithm, the neural network models implementing the boosting algorithm were shown to produce networks with less bias.

4.5 Conclusions

The study investigated the utility of neural network models in predicting the saturated hydraulic conductivity at field scale. Two different case-studies with different soil types were considered for the analysis. Two different ensemble neural network models, one using bagging algorithm and the other using boosting algorithm were developed and tested on the two case-studies. The performance of the field-scale artificial neural network models were compared with the published neural network program, *Rosetta*.

For both the case-studies, the field-scale neural network models performed better than the *Rosetta* model. This emphasizes that a neural network model trained even on a small set of relevant data, when available, is better than training a neural network model with a large but more general data set. For both the field-scale models, the inclusion of ρ_b as one of the inputs to the neural networks increased the uncertainty in the model predictions.

In contrast to most of the earlier studies that employed bagging algorithm to improve the performance of the neural network models in predicting the soil hydraulic properties, the study demonstrated the superior performance of the boosting algorithm based ensemble networks in modeling the saturated hydraulic conductivity at field scale. The Field(Boosting) model consistently resulted in less mean residual values than the Field(Bagging) model indicating the lower bias associated with the latter model. Compared to the bagging algorithm, the boosting algorithm reduced both the bias and variance of the neural network models. The utility of the boosting algorithm in improving the performance of the neural network models with regards to modeling soil hydraulic parameters needs to be further explored on large scale databases.

4.6 Acknowledgements

Fundings for this project were provided by the National Science and Engineering Research Council of Canada (NSERC) to AE and BCS. Technical help from W. Bodhinayake, T. Zeleke and L. Tallon is greatly appreciated.

4.7 References

Anderson, D.W., and Ellis, J.G. (1976). The soils of provincial forest reserves in the Prince Albert map area – 73H Saskatchewan. Saskatchewan Institute of Pedology, University of Saskatchewan, Saskatoon, Sask.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. (2000a). "Artificial neural networks in hydrology I: Preliminary concepts." *J. Hydrol. Eng.,* 5, 115-123.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. (2000b). "Artificial neural networks in hydrology II: Hydrologic application." *J. Hydrol. Eng.,* 5, 124-137.

Bastet, G., Bruand, A., Voltz, M., Bornand, M., and Quetin, P. (1999). Performance of available pedotransfer functions for predicting the water retention properties of French soils. p. 981-991. In M.Th. Van Genuchten et al. (ed.) *Characterzation and Measurement of the Hydraulic Properties of Unsaturated Porous Media*. Univ. of California, Riverside, CA.

Bouma, J. (1989). "Using soil survey data for quantitative land evaluation." *Adv. Soil Sci*., 9, 177-213.

Breiman, L. (1996). "Bagging predictors." *Mach. Learn.,* 24, 123-140.

Brown, M., and Harris, C. (1994). *Neurofuzzy adaptive modeling and control*. Prentice Hall: New York.

Cosby, B. J., Hornberger, G. M., Clapp, R. B., and Ginn, T. R. (1984). "A statistical expolaration of the relationships of soil moisture characteristics to the physical properties of soils." *Water Resour. Res.,* 20, 682-690.

Demuth, H., and Beale, M. (2001). *Neural network toolbox learning. For use with MATLAB*. The Math Works Inc, MA.

Drucker, H. (1999). Boosting using neural networks.P. 51-78. In A.J.C. Sharkey, (ed.) *Combining artificial neural nets*. Springer-Verlag, London.

Efron, B., and Tibshirani, T. J. (1993). *An Introduction to Bootstrap*. Chapman and Hall, New York, NY.

Freund, Y., and Schapire, R. E. (1996). Experiments with a new Boosting algorithm. P. 148-156. In L. Saitta (ed.) Proceedings of the Thirteenth International Conference on Machine Learning (ICML), Bari, Italy, 3-6 July 1996. Morgan Kaufmann, San Francisco, CA.

Gee, G. W., and Bauder, J. W. (1986). Particle-size analyses, p. 384-423. In A. Klute (ed.) *Method of soil analyses*. *Part 1*. Agron. Monogr. No. 9, ASA and SSSA, Madison, WI.

Haykin, S. S. (1999). *Neural networks: A comprehensive foundation*. Prentice Hall, NJ.

Hsu, K., Gupta, V. H., and Sorooshian, S. (1995). "Artificial neural network modeling of the rainfall-runoff process." *Water Resour. Res*., 31(10), 2517-2530.

Klute, A., and Dirksen, C. (1986). Hydraulic conductivity and diffusivity: Laboratory methods, p. 687-734. In A. Klute (ed.) *Method of soil analyses, Part 1*, Agron. Monogr. No. 9, ASA and SSSA, Madison, WI.

Legates, D. R., and McCabe, Jr. G. J. (1999). "Evaluating the use of goodness-of-fit measures in hydrologic and hydroclimatic model validation." *Water Resour. Res.,* 35, 233-241.

Leij, F., Schaap, M. G., and Arya, L. M. (2002). Water retention and storage: Indirect methods. p. 1009-1045. *In* J.H. Dane and G.C. Topp (ed.) *Methods of soil analysis: Part 4.* SSSA Book Ser. No. 5. SSSA, Madison, WI.

Maier, H. R., and Dandy, G. C. (2000). "Neural networks for the prediction and forecasting of water resources variables: a review of modeling issues and application." *Environ. Modell. Software,* 15, 101-124.

Minasny, B., and McBratney, A. B. (2002). "The neuro-m method for fitting neural network parametric pedotransfer functions." *Soil Sci. Soc. Am. J.,* 66, 352-361.

Minasny, B., Hopmans J. W., Harter, T., Eching, S. O., Tuli, A., and Denton, M. A. (2004). "Neural networks prediction of soil hydraulic functions for alluvial soils using multistep outflow method." *Soil Sci. Soc. Am. J.,* 68, 417-429.

Minasny, B., McBratney, A. B., and Bristow, K. L. (1999). "Comparison of different approaches to the development of pedotransfer functions for water retention curves." *Geoderma,* 93, 225-253.

Nemes, A., Schaap, M. G., and Wösten, J. H. M. (2003). "Functional evaluation of pedotransfer functions derived from different scales of data collection." *Soil Sci. Soc. Am. J.,* 67, 1093-1102.

Pachepsky, Y. A., Timlin, D. J., and Varallyay, G. (1996). "Artificial neural networks to estimate soil water retention from easily measurable data." *Soil Sci. Soc. Am. J.,* 60, 727-773.

Pandey, G. R., and Nguyen, V. T. V. (1999). "A comparative analysis of regression based methods in regional flood frequency analysis." *J. Hydrol.,* 225, 92-101.

Rawls, W. J., and Brakensiek, D. L. (1983). A procedure to predict Green and Ampt infiltration parameters. P. 102-112. In *Adv. in Infiltration*. ASAE, St. Joseph, MI.

Rawls, W. J., Gish, T. J., and Brakensiek, D. L. (1991). "Estimating soil water retention from soil physical properties and characteristics." *Adv. Soil Sci.,* 9, 213-234.

Romano, N., and Palladino, M. (2002). "Prediction of soil water retention using soil physical data and terrain attributes." *J. Hydrol.,* 265, 56-75.

Saxton, K. E., Rawls, W. J., Romberger, J. S., and Papendick, R. I. (1986). "Estimating generalized soil-water characteristics from texture." *Soil Sci. Soc. Am. J.,* 50, 1031-1036.

Schaap, M. G., and Bouten, W. (1996). "Modeling water retention curves of sandy soils using neural networks." *Water Resour. Res.,* 32(10), 3033-3040.

Schaap, M.G., Leij, F. L., and Van Genuchten M. Th. (1998). "Neural network analysis for hierarchical prediction of soil hydraulic properties." *Soil Sci. Soc. Am. J.,* 62, 847-855.

Schaap, M. G., Leij, F. L., and Van Genuchten, M. Th. (2001). "Rosetta: A computer program for estimating soil hydraulic parameters with hierarchical pedotransfer functions." *J. Hydrol.,* 251, 163-176.

Schapire, R. E. (1990). "The strength of weak learnability." *Mach. Learn.,* 5, 197-227.

Shu, C., and Burn, D. H. (2004). "Artificial neural network ensembles and their application in pooled flood frequency analysis." *Water Resour. Res.,* 40, W09301, doi:10.1029/2003WR002816.

Si, B. C., and Farrell, R. E. (2004). "Scale-dependent relationship between wheat yield and topographic indices: A wavelet approach." *Soil Sci. Soc. Am. J.,* 68, 577-587.

Si, B. C., and Zeleke, T. B. (2005). "Wavelet coherency to relate soil saturated hydraulic conductivity and physical properties." *Water Resour. Res.,* 41, W11424, doi:10.1029/2005WR004118.

Tamarai, S., and Wösten, J. H. M. (1999). Using artificial neural networks to develop pedotransfer functions of soil hydraulic conductivity. p. 1251-1260. In M.Th. Van Genuchten et al. (ed.) *Characterzation and Measurement of the Hydraulic Properties of Unsaturated Porous Media*. Univ. of California, Riverside, CA.

Tietje, O., and Tapkenhinrichs, M. (1993). "Evaluation of pedotransfer functions." *Soil Sci. Soc. Am. J.,* 57, 1088-1095.

Tietje, O., and Hennings, V. (1996). "Accuracy of the saturated hydraulic conductivity prediction by pedo-transfer functions compared to the variability within FAO textural classes." *Geoderma,* 69, 71-84.

van Genuchten, M.Th., Leij, F. J., and Lund. L. J. (1992). On estimating the hydraulic properties of unsaturated soils. P. 1-14. In M.Th. van Genuchten et al. (ed*.) Indirect methods for estimating the hydraulic properties of unsaturated soils*. Proc. Int. Workshop. Riverside, CA. 11-13 Oct. 1989. Univ. of California, Riverside.

Vereecken, H., Maes, J., and Feyen, J. (1990). "Estimating unsaturated hydraulic conductivity from easily measured soil properties." *Soil Sci.,* 149, 1-12.

Wösten, J. H. M. (1990). "Use of soil survey data to improve simulation of water movement in soils." Ph. D. thesis. Univ. of Wageningen, the Netherlands.

Wösten, J. H. M., Finke, P. A., and Jansen, M. J. W. (1995). "Comparison of class and continuous pedotransfer functions to generate soil hydraulic characteristics." *Geoderma,* 66, 227-237.

Wösten, J. H. M., Lilly, A., Nemes, A., and Le Bas, C. (1999). "Development and use of a database of hydraulic properties of European soils." *Geoderma,* 90, 169-185.

Wösten, J. H. M., Pachepsky, Y. A., and Rawls, W. J. (2001). "Pedotransfer functions: Bridging gap between available basic soil data and missing soil hydraulic characteristics." *J. Hydrol.,* 251, 123-150.

Zeleke, T. B., and Si, B. C. (2005). "Scaling Relationships between Saturated Hydraulic Conductivity and Soil Physical Properties." *Soil Sci. Soc. Am. J.,* 69, 1691-1702.

Table 4.1 Statistics of the entire dataset along with the dataset used for training and testing (Smeaton)

Variable	Units	Entire Dataset (N=126)					Training (N=84)					Testing (N=42)				
		Min.	Max.	Avg.	SD	CV	Min.	Max.	Avg.	SD	CV	Min.	Max.	Avg.	SD	CV
Sand	$g\ g^{-1}\times100$	52.50	86.30	65.00	7.30	0.11	53.80	86.30	65.90	7.60	0.12	52.50	76.30	63.20	6.29	0.10
Silt	$g\ g^{-1}\times100$	13.30	42.50	30.30	6.30	0.21	13.30	42.30	29.50	6.40	0.22	20.00	42.50	31.70	5.73	0.18
Clay	$g\ g^{-1}\times100$	0.50	15.00	4.90	2.70	0.55	0.50	15.00	4.80	2.80	0.58	1.30	11.30	5.20	2.51	0.48
BD	$g\ cm^{-3}$	1.10	1.52	1.32	0.09	0.07	1.13	1.52	1.32	0.09	0.07	1.10	1.46	1.31	0.09	0.07
$\log_{10}(K_s)$	$\log_{10}$ (cm day^{-1})	1.13	2.29	1.86	0.26	0.14	1.13	2.29	1.84	0.28	0.15	1.49	2.29	1.91	0.21	0.11

Table 4.2 Statistics of the entire dataset along with the dataset used for training and testing (Alvena)

Variable	Units	Entire Dataset (N=78)					Training (N=52)					Testing (N=26)				
		Min.	Max.	Avg.	SD	CV	Min.	Max.	Avg.	SD	CV	Min.	Max.	Avg.	SD	CV
Sand	$g\ g^{-1}\times100$	9.00	24.00	16.00	4.04	0.25	9.00	24.00	16.00	4.06	0.25	10.00	24.00	16.00	4.09	0.26
Silt	$g\ g^{-1}\times100$	46.00	63.00	55.00	4.18	0.08	47.00	63.00	55.00	4.10	0.07	46.00	62.00	55.00	4.43	0.08
Clay	$g\ g^{-1}\times100$	20.00	41.00	29.00	4.99	0.17	20.00	41.00	29.00	5.21	0.18	23.00	39.00	29.00	4.62	0.16
BD	$g\ cm^{-3}$	1.11	2.35	1.33	0.17	0.13	1.11	2.35	1.33	0.19	0.14	1.14	1.57	1.32	0.12	0.09
$\log_{10}(K_s)$	$\log_{10}$ (cm day^{-1})	0.74	2.90	1.88	0.46	0.24	0.90	2.90	1.91	0.44	0.23	0.74	2.62	1.82	0.49	0.27

Table 4-3 Performance statistics of different models on the Smeaton dataset

Model	Training				Testing				Rank Score
	RMSE	MRE	MR	Uncertainty	RMSE	MRE	MR	Uncertainty	
				SSC					
Rosetta	0.27	0.12	0.06	0.15	0.26	0.11	0.15	0.16	0.00
Field(Bagging)	0.23	0.10	-0.02	0.04	0.23	0.09	0.08	0.04	0.75
Field(Boosting)	0.23	0.11	-0.03	0.04	0.22	0.09	0.07	0.04	1.00
				SSC+ρ_b					
Rosetta	0.29	0.14	-0.17	0.12	0.19	0.08	-0.07	0.12	0.63
Field(Bagging)	0.21	0.10	-0.01	0.11	0.20	0.08	0.07	0.11	0.63
Field(Boosting)	0.22	0.10	-0.05	0.10	0.20	0.08	0.04	0.10	0.88

Table 4-4 Performance statistics of different models on the Alvena dataset

Model	Training				Testing				Rank Score
	RMSE	MRE	MR	Uncertainty	RMSE	MRE	MR	Uncertainty	
				SSC					
Rosetta	0.92	0.40	0.81	0.11	0.87	0.38	0.72	0.10	0.13
Field(Bagging)	0.38	0.20	-0.09	0.10	0.48	0.28	-0.19	0.09	0.88
Field(Boosting)	0.38	0.20	-0.08	0.12	0.48	0.28	-0.17	0.10	0.88
				SSC+ρ_b					
Rosetta	0.96	0.38	0.70	0.13	0.86	0.42	0.60	0.12	0.00
Field(Bagging)	0.37	0.19	-0.13	0.12	0.50	0.29	-0.23	0.11	0.63
Field(Boosting)	0.36	0.18	-0.09	0.11	0.48	0.28	-0.19	0.11	1.00

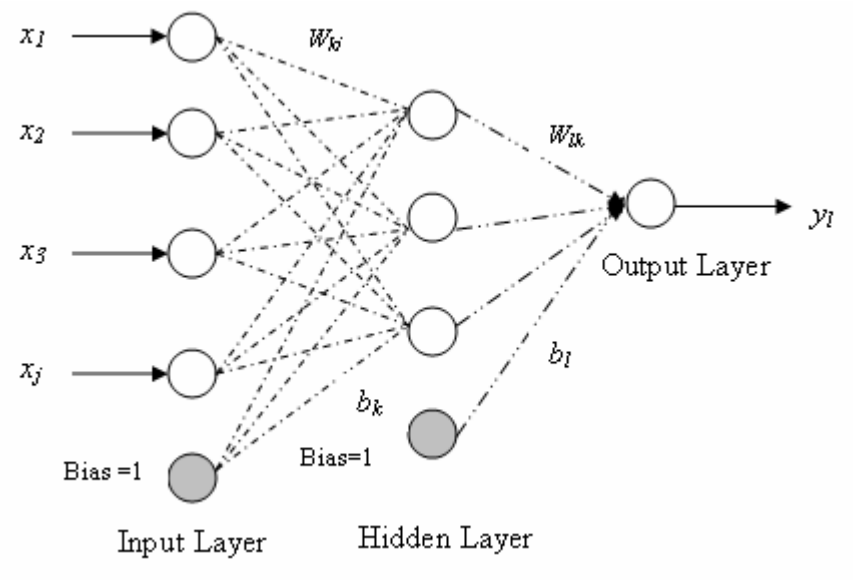

Figure 4-1. Structure of the three-layered feed-forward neural network (FF-NN).

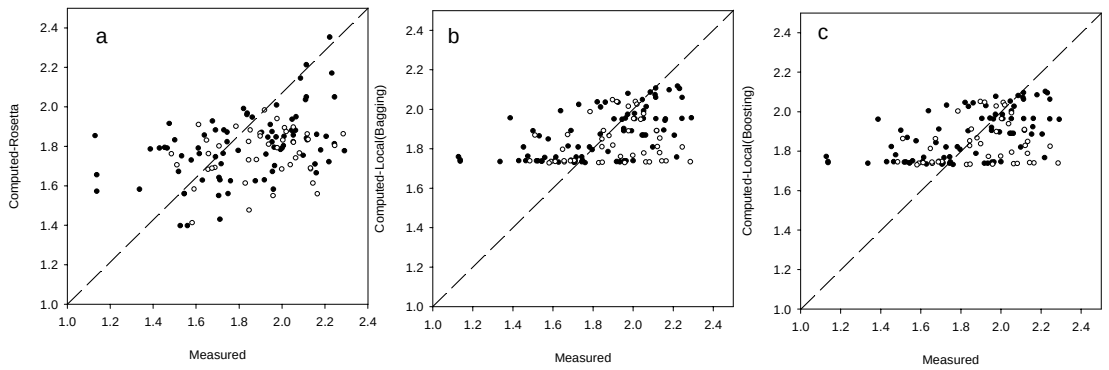

Figure 4-2 Scatter plots between the measured and the computed $\log_{10}$(Ks) by (a) Rosetta; (b) Field(Bagging); and (c) Field(Boosting) for Smeaton with SSC as inputs. The 'solid' points represent the training instances and the 'open' points represent the testing instances.

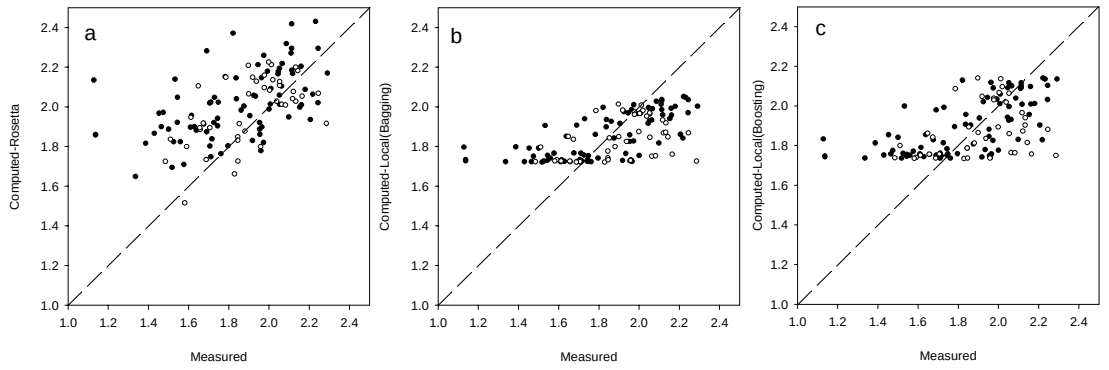

Figure 4-3 Scatter plots between the measured and the computed $\log_{10}$(Ks) by (a) Rosetta; (b) Field(Bagging); and (c) Field(Boosting) for Smeaton with SSC and ρ_b as inputs. The 'solid' points represent the training instances and the 'open' points represent the testing instances.

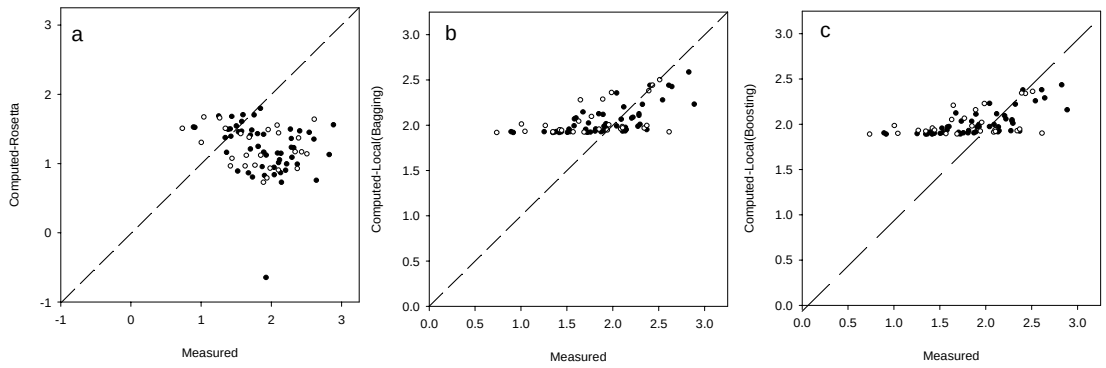

Figure 4-4 Scatter plots between the measured and the computed $\log_{10}$(Ks) by (a) Rosetta; (b) Field(Bagging); and (c) Field(Boosting) for Alvena with SSC as inputs. The 'solid' points represent the training instances and the 'open' points represent the testing instances.

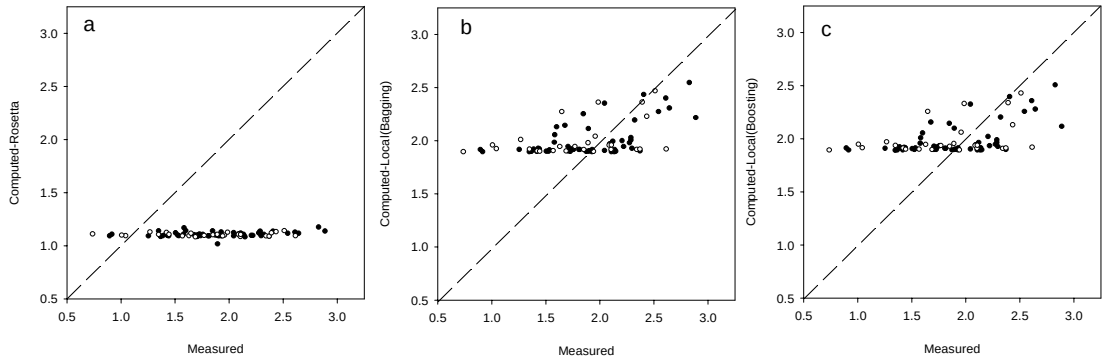

Figure 4-5 Scatter plots between the measured and the computed $\log_{10}$(Ks) by (a) Rosetta; (b) Field(Bagging); and (c) Field(Boosting) for Alvena with SSC and ρ_b as inputs. The 'solid' points represent the training instances and the 'open' points represent the testing instances.

Chapter 5 - Estimating Saturated Hydraulic Conductivity Using Genetic Programming

Citation: Parasuraman, K., Elshorbagy, A., and Si, B. C. (2007). "Estimating saturated hydraulic conductivity using genetic programming." *Soil Sci. Soc. Am. J.,* Accepted (May 11, 2007).

Contribution of the PhD candidate

Model conceptualization was carried out by Kamban Parasuraman and Dr. Amin Elshorbagy. Model development was carried out by Kamban Parasuraman, with Dr. Amin Elshorbagy and Dr. Bing Cheng Si providing guidance at various stages of the work. Dr. Bing Cheng Si helped in obtaining the dataset used in this study. The text of the published paper was created by Kamban Parasuraman with Dr. Amin Elshorbagy critically reviewing the manuscript.

Contribution of this chapter to the overall study

This chapter is a continuation of the previous chapter, where the objective was to improve the reliability of the system-theoretic models. In this chapter, a methodology for improving the reliability of geophysical models by accounting for the influence of model-structure uncertainty is proposed. The methodology was applied to develop pedotransfer functions for estimating saturated hydraulic conductivity of soils. The uncertainty due to model structure is shown to be more than the uncertainty due to model parameters. `An

increase in the model complexity is shown to increase the predictive ability of the model, but at an increasing level of uncertainty.

5.1 Abstract

Saturated hydraulic conductivity (K_s) is one of the key parameters in modeling the solute and water movement in the vadose zone. Field and laboratory measurement of K_s is time consuming, and hence is not practical for characterizing the large spatial and temporal variability of K_s. As an alternative to direct measurements, pedotransfer functions (PTFs), which estimate K_s from readily available soil data are being widely adopted. This study explores the utility of a promising data-driven method, namely, genetic programming (GP), to develop PTFs, for estimating K_s from sand, silt, clay contents, and bulk density (BD). A dataset from the UNsaturated SOil hydraulic DAtabase (UNSODA) has been considered in this study. The performances of the GP models are compared with the neural networks (NNs) model, as it is the most widely adopted method for developing PTFs. The uncertainty of the PTFs is evaluated by combining the GP and the NN models, with the non-parametric bootstrap method. Results from the study indicate that GP appears to be a promising tool for developing PTFs for estimating K_s. The better performance of the GP model may be attributed to the ability of GP to optimize both the model structure and its parameters in unison. For the PTFs developed using GP, the uncertainty due to model structure is shown to be more than the uncertainty due to model parameters. An increase in the model complexity is shown to increase the predictive ability of the model, but at an increasing level of uncertainty.

5.2 Introduction

Over the past few decades, vadose zone modeling has received significant impetus due to the advancement in computing power and technology. Hence, focus on vadose zone models has shifted from coarse lumped models to more realistic spatially distributed models. These spatially distributed models have drastically increased the need for soil hydraulic data on finer resolution. Direct (field and laboratory) measurement of soil hydraulic data is labour intensive, time consuming and expensive as these methods require restrictive initial and boundary conditions. For a detailed review of different laboratory and field measurement of soil hydraulic data, readers are referred to Klute (1986). The problems (labor intensive, time consuming, expensive) associated with direct measurement of soil hydraulic properties make it quite impractical to amass a dataset at a resolution required for implementing such a spatially distributed model. Alternatively, these soil hydraulic properties can be estimated from more easily available soil data by the use of pedotransfer functions (PTFs) (Bouma, 1989).

PTFs are predictive functions that can translate basic soil data like particle-size distributions, bulk density, and organic matter content, into soil hydraulic properties. Due to this reason, interest in developing PTFs is ever increasing (Rawls and Brakensiek, 1983; Cosby et al., 1984; Saxton et al., 1986; Vereecken et al., 1990; Van Genuchten et al., 1992; and Leij et al., 2002). A detailed review of different PTFs is given by Wösten et al. (2001). Several methods have been adopted in the literature to develop PTFs. These methods range from simple lookup tables to more complex data-driven methods like regression analysis, neural networks (NN), group method of data handling, and regression

trees. Gupta and Larson (1979) used linear regression to estimate soil water characteristic. Rawls et al. (1991) and Minasny et al. (1999) used nonlinear regression to develop PTFs. The regression models are being gradually replaced by the NN models in developing PTFs. Key examples of such studies include Pachepsky et al. (1996), Schaap and Bouten (1996), Minasny et al. (1999), and Tamarai et al. (1998). Another data-driven technique, called the group method of data handling has been used by Pachepsky et al. (1998), Nemes et al. (2005) and Ungaro et al. (2005) for developing PTFs. The technique of regression trees has been used by McKenzie and Jacquier (1997) for developing PTFs. As evident from a plethora of studies in literature on NN based PTFs, NN appears to be the most widely adopted method for developing PTFs.

Recently, another promising inductive data-driven technique called Genetic programming (GP) was proposed by Koza (1992). GP is a method for constructing populations of models using stochastic search methods namely evolutionary algorithms. An important characteristic of GP is that, both the variables and constants of the candidate models are optimized. Hence, compared to other regression techniques, it is not required to choose the model structure a priori. In water related studies, GP has been applied to model: flow over a flexible bed (Babovic and Abbott, 1997); rainfall-runoff process (Whigham and Crapper, 2001; Savic et al., 1999); runoff forecasting (Khu et al., 2001; urban fractured-rock aquifer dynamics (Hong and Rosen, 2002); temperature downscaling (Coulibaly, 2004); rainfall-recharge process (Hong et al., 2005); soil moisture (Makkeasorn et al., 2006); and evapotranspiration (Parasuraman et al., 2007).

Although GP and the most widely used method for developing PTFs, namely NN, can be seen as alternative techniques for the same task, like, e.g., classification and approximation problems, in contrast to NN, studies to determine the utility of GP in developing PTFs has not been attempted yet. Hence in this study, an attempt has been made to explore the efficacy of GP in developing PTFs for estimating the saturated hydraulic conductivity (K_s). Specific objectives of this study include (i) developing PTFs for estimating K_s using GP; (ii) comparing the performance of the GP-based PTFs with the performance of the NN-based model, as it is the most widely used method for developing PTFs; and (iii) highlighting the potential as well as the shortcomings of the use of GP for geophysical applications.

5.3 Materials and Methods

5.3.1 Dataset Used

The dataset used in this study is derived from the UNsaturated SOil hydraulic DAtabase (UNSODA) database (Leij et al., 1996). The UNSODA database was developed to provide a source of hydraulic data and other soil properties for practitioners and researchers, and the database was derived from soil samples from Europe and North America. Compared to soils with finer texture, coarse textured soils are in a majority in the UNSODA database (Leij et al., 1996). The UNSODA database has been widely used for developing PTFs (e.g., Schaap and Leij, 1998; Schaap et al., 2001; Ungaro et al., 2005). Sand, silt, clay content (SSC), bulk density (BD) and K_s values of 314 samples are extracted from the UNSODA database. Out of the 314 samples, 200 and 114 samples

were selected for model calibration and validation, respectively. The K_s values are log-transformed to account for their log-normal distribution. One of the samples in the calibration dataset had K_s=1 cm/day, which when log-transformed results in $\log_{10}(K_s)$=0. As detailed in later section, mean absolute relative error (MARE) is used as one of the measures for evaluating the models performance. Hence, the above mentioned value is discarded from the calibration set, as it is not possible to calculate the relative error for that particular sample. As the UNSODA database is created by assembling different sources of data that came different parts of the world, the dataset as such is stratified randomly. Hence, no special consideration was given to specifically sample the dataset between the training and the testing set. From the available 313 data instances, the first 199 instances were selected for training, and the remaining 114 data instances were used for testing. The descriptive statistics, along with the correlation matrix, of the dataset used for calibration and validation are presented in Table 5-1 and Table 5-2, respectively. The coefficient of variation (CV) of different variables during training and testing are comparable (Table 5-1 and Table 5-2).

5.3.2 Genetic Programming

Genetic Programming (GP), introduced by Koza (1992), is a new addition to a class of evolutionary algorithms like Evolutionary Programming (Fogel et al., 1966), Genetic Algorithms (Holland, 1975), and Evolution Strategies (Schwefel, 1981). In GP, a population of solution candidates evolves through many generations towards an optimal solution, using the concepts of natural selection and genetics. Genetic symbolic regression (GSR) is a special application of GP in the area of symbolic regression, where

the objective is to find a mathematical expression in symbolic form, that provides an optimal fit between a finite sampling of values of the independent variable and its associated values of the dependent variable (Koza, 1992).

GSR can be considered as an extension of numerical regression problems. In numerical regression problems, one predetermines the functional form (linear, quadratic, or polynomial), and the objective is to find the set of numerical coefficients that best fits the chosen model structure. However, GSR does not require the functional form to be defined a priori, as GSR involves finding the optimal mathematical expression in symbolic form (both the discovery of the correct functional form and the appropriate numerical coefficients) that defines the predictand-predictor relationship. More information on GP can be found in Koza (1992) and Babovic and Keijzer (2000).

Figure 5-1 shows the flowchart of the GSR paradigm. For a given problem, the first step is to define the functional and terminal set, along with the objective function and the genetic operators. Functional set and terminal set are the main building blocks of the GSR, and hence their appropriate identification is central in developing a robust GSR model. The functional set consists of basic mathematical operators { +, -, *, /, sin, exp, …} that may be used to form the model. The choice of operators depends upon the degree of complexity of the problem to be modeled. The terminal set consists of independent variables and constants. The constants can either be physical constants (e.g. Earth's gravitational acceleration, specific gravity of fluid) or randomly generated constants. Different combinations of functional and terminal sets are used to construct a

population of mathematical models. Each model (individual) in the population can be considered as a potential solution to the problem. The mathematical models are usually coded in a parse tree form. For example, Figure 5-2 shows the parse tree notation of a mathematical model $f(x, y, z) = (x + y) * (6/z)$. In Figure 5-2, the connection points are called as nodes, and it can be seen that the inner nodes of the parse tree are made up of functions, and the terminal nodes are made up of variables and constants. Hence in GP terminology, the variables and constants are referred as terminals, and the functions are referred as non-terminals. The depth of the sparse tree shown in Figure 5-2 is three. Objective or fitness or cost function is used to evaluate the value or fitness of each individual in the population. Usually squared error statistics or its variant (mean squared error and root mean squared error) is used as the objective function. Genetic operators include selection, crossover, and mutation, and they are discussed in detail later in this section.

Once the functional and terminal sets are defined, the next step is to generate the initial population for a given population size. The initial population can be generated in a multitude of ways, including, the full method, grow method, and ramped half-and-half method. In the full method, the new trees are generated by assigning non-terminal nodes until a pre-described initial maximum tree depth is reached, and the last depth level is limited to the terminal node. The full method usually results in perfectly balanced trees with branches of same length. In the grow method, each new node is randomly chosen between the terminals and the non-terminals, with the terminals making up the node at the initial maximum tree depth. As a consequence, the grow method usually results in

highly unbalanced trees. The ramped half-and-half method is a combination of the full and the grow methods. For each depth level considered, half of the individuals are initialized using the full method and the other half using the grow method. The ramped half-and-half method is shown to produce highly diverse trees, both in terms of size and shape (Koza, 1992), and thereby provides a good coverage of the search space. More information on the different methods of generating the initial population can be found in Koza (1992). Once initialized, the fitness of each individual (mathematical model) in the population is evaluated based on the selected objective function. The higher the fitness of an individual, the greater is the chance of the individual being carried over to the next generation. At each generation, new sets of models are evolved by applying the genetic operators: selection, crossover and mutation (Koza, 1992; Babovic and Keijzer, 2000). These new models are termed offspring, and they form the basis for the next generation.

After the fitness of the individual models in the population is evaluated, the next step is to carry out selection. The objective of the selection process is to create a temporary population called the mating pool, which can be acted upon by genetic operators, crossover and mutation. Selection can be carried out by several methods like truncation selection, tournament selection, and roulette wheel selection (Koza, 1992). As roulette wheel selection is the most widely used method including Koza (1992), it has been adopted in this study. Roulette wheel is constructed by proportioning the space in the roulette wheel based on the fitness of each model in the population. The selection process ensures that the models with higher fitness have more chance of being carried over to the next generation.

Crossover is carried out by choosing two parent models from the mating pool and swapping corresponding sub-tree structures across a randomly chosen point to produce two different offspring with different characteristics. The number of models undergoing crossover depends upon the chosen probability of crossover, P_c. Mutation involves random alteration of the parse tree at the branch or node level. This alteration is done based on the probability of mutation, P_m. For an overview of different types of computational mutations, readers are referred to Babovic and Keijzer (2000). While the role of crossover operator is to generate new models, which did not exist in the old population, the mutation operator guards against premature convergence by constantly introducing new offspring into the population. Figure 5-3 demonstrates the crossover and mutation operators. The crossover point between Parent 1 and Parent 2 is shown by the dashed line, and the corresponding sub-tree structures are swapped, resulting in Offspring 1 and Offspring 2. In Offspring 1, the terminal node has undergone mutation (2 replaced by z). The genetic operators, crossover and mutation, are shown to produce new models (Offspring), which are structurally different from their parent models (Figure 5-3). These operators ensure that the model space is sampled thoroughly to arrive at the optimal model. After the initial population has been acted upon by the genetic operators, the resultant individuals form the new population for the next generation. This iterative process is carried out for a predetermined number of iterations or until a specified value of cost function is reached.

In this study, the GP system used is an adaptation of GPLAB (Silva, 2005), a GP toolbox for MATLAB. Since the values of the GP system parameters (e.g. crossover rate, mutation rate, population size) are problem dependent, the usual practice is to determine them using trial-and-error process with the objective of minimizing the cost function during the training process. This study adopted a similar approach in arriving at the GP parameters, and the resulting parameter values are shown in Table 5-3. One of the main issues that need to be addressed in developing a GP system is that of "bloating". Bloating refers to the exponential growth of redundant and functionally useless trees. This is caused by the genetic operators (crossover and mutation) in their quest to arrive at better solutions. Several bloat control techniques have been proposed in the literature, and a review of these methods can be found in Soule and Foster (1999), Poli (2003), and Silva and Costa (2004). This study adopted the Heavy Dynamic Limit method proposed by Silva and Costa (2004), which is based on attaching a dynamic limit on the depth of the trees allowed in the population, initially set with a low value, and only raised and lowered when needed to accommodate an individual with better performance that would otherwise break the limit. More information on the heavy dynamic limit method can be found in Silva and Costa (2004).

Two GP models, GP(1) and GP(2), are developed to estimate K_s. While the functional set of GP(1) and GP(2) models are { + ,- } and { + ,-, /, * } respectively, the terminal set of both the models remains the same. Along with randomly generated constants, *{Sand, Silt, Clay, Bulk density}* constitute the terminal set of both GP(1) and GP(2) models. This exercise is carried out to evaluate the performance of the GP models

with varying level of mathematical operators (complexity) that can be used to define the predictand-predictor relationship. Prior to developing PTFs for estimating K_s using GP, both the independent (Sand, Silt, Clay, BD) and the dependent variables ($\log_{10}(K_s)$) are normalized by dividing each variable by their corresponding maximum value. This is done in order to overcome the problem of dimensional inconsistency and to achieve better generalization. These standardized values, herein are simply referred as Sand, Silt, Clay, BD and K_s.

5.3.3 Performance Evaluation

The performances of the GP-based models are compared with the NN model, as it is the most widely used method for developing PTFs. A detailed description of NN is beyond the scope of this paper. For a detailed understanding of NNs, readers are referred to Haykin (1999). The NN model adopted in this study employs Bayesian-Regularization (BR) algorithm for training the networks. The BR algorithm has the advantage of producing networks with good generalization property as the cost function (Equation 5.1) involves minimizing both the mean sum of squares of the network errors (MSE) and the mean of the sum of squares of the network weights and bias (MSW). In Equation (5.1), y_i and $y_i^{'}$ represent the measured and computed $\log_{10}(K_s)$ values; α represents the regularization parameter; w_j represents the connection weights and bias values; n, and N represents the number of training instances and the number of network parameters respectively. More information on BR algorithms can be found in Demuth and Beale (2001). By trial-and-error method, the optimal number of hidden neurons was found to be six. The hidden layer neurons uses tan-sigmoidal activation function and the output layer

neurons use a linear activation function. Herein, the NN model will be referred as NN(BR). The performances of the different models are evaluated based on (i) root mean squared error (RMSE), (ii) mean absolute relative error (MARE), and (iii) mean residual (MR). RMSE, MARE, and MR statistics are calculated using Equations (5.2), (5.3), and (5.4) respectively. In Equations (5.2), (5.3), and (5.4), similar to Equation (5.1), y_i and $y_i^{'}$ represents the measured and computed $\log_{10}(K_s)$ values, and n represents the number of data instances.

$$MSE\_REG = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^{'}\right)^2 + (1-\alpha)\left(\frac{1}{N}\sum_{j=1}^{N} w_j^2\right) \tag{5.1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^{'})^2} \tag{5.2}$$

$$MARE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - y_i^{'}}{y_i}\right| \tag{5.3}$$

$$MR = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i^{'}\right) \tag{5.4}$$

5.3.4 PTF Uncertainty

Wösten (1990) and McBratney et al. (2002) underscore the need for uncertainty estimates as part of the PTF performance evaluation. The uncertainty estimate, unlike the RMSE, MARE, and MR statistics, indicates the reliability of K_s estimated by the model. Calculation of RMSE, MARE, and MR is possible only when measured K_s values are available. In such instances, uncertainty estimate can still provide the measure of reliability of K_s estimated by the model. Usually, PTF uncertainty is calculated using the

non-parametric bootstrap method (Efron and Tibshirani, 1993). The bootstrap method presumes that the training dataset is a good representation of the original population, and that this dataset is only one particular realization of that population. Hence, training the model on a different realization of the population would result in a slightly different prediction of K_s. In order to account for such uncertainty in prediction, B independent datasets (T_B), of size N, can be generated by repeated random resampling with replacement of the training dataset (T), of size N. Hence each bootstrap dataset T_B may have many instances of T repeated several times, while other instances may be left out. Since T_B contains a different realization of T, models trained on each of T_B may be slightly different. It should be noted that the different realization produced by the bootstrap method are different not in terms of relative values, but only in the order and occurrence of the data instances. For this study, bootstrap size, B, is assumed to be 50. For a rational comparison, initially the resampled datasets (T_B) are generated and predetermined so that the NN and GP models can be trained on the same resampled datasets. The model accuracy and its related uncertainty is calculated in the following manner: (i) since a bootstrap size of 50 is used in this study, for each input vector in the dataset, the NN- and GP- PTFs results in 50 different model predictions, based on models trained on the 50 resampled datasets (i.e. for each input vector the PTF results in 50 different model predictions); (ii) for that particular input vector, the estimated PTF value and its related uncertainty is determined by calculating the mean and standard deviations of the 50 different model predictions. The mean represents the model estimated value, and the standard deviation represents the uncertainty associated with that particular model estimate; (iii) similarly, the model estimates and their related uncertainty are

evaluated for all the input vectors in the training and the testing datasets; (iv) for a particular PTF, the performance in terms of RMSE, MARE, and MR statistics is then calculated by comparing the model estimates with their measured counterparts over the entire training and testing ranges; (v) the overall uncertainty associated with that particular PTF is calculated by averaging the standard deviations of the ensemble model predictions over the entire training and testing ranges.

Adopting ensemble technique in PTFs development not only assists in evaluating the uncertainty of the developed PTFs, but also helps in addressing one of the pertinent issues in any machine learning (e.g. ANNs, GP) algorithms, namely generalization. Iba (1999) applied the ensemble method of bagging and boosting within the framework of GP and obtained encouraging results. Keijzer and Babovic (2000) and Folino et al. (2006) demonstrated that ensemble methods like bagging and boosting can reduce the generalization error in GP. Hence, the models developed in this study by combining self-organizing algorithm with statistical resampling techniques are expected to reduce, if not fully overcome, the generalization error. It should be noted that in the case of the GP models (GP(1) and GP(2)), for each T_B, both the model structure and the model parameters are evolved simultaneously by the self-organizing nature of the GP algorithm. Nevertheless in NN(BR) model, for each of T_B, the model structure is assumed to be deterministic, with the model parameters alone optimized based on T_B. Although, the framework of the GP and NN(BR) models are not functionally identical, the comparison of the above models is effected in order to illustrate the value of self-

organizing ability of the GP-based PTFs, proposed in this study, against the conventional way by which NN-based PTFs were developed in literature.

5.4 Results and Analysis

The performance statistics of different models in estimating K_s, during both training and testing, are presented in Table 5-4. During training, NN(BR) resulted in an RMSE of 0.61, MARE of 0.55, MR of -0.01, and an uncertainty of 0.26 (Table 5-4). However, during testing, the corresponding values were 1.04, 2.23, -0.09, and 0.27, respectively (Table 5-4). It can be observed that, compared to training, the testing RMSE and MARE estimates increased significantly in the case of NN(BR) model as compared to the GP models (Table 5-4). This demonstrates that although the NN(BR) model was robust in learning the input-output patterns in the training dataset, it was not able to generalize the relationship.

During training, GP(2) (more complex) model performed better than the GP(1) (less complex) model in terms of RMSE and MARE; and equally in terms of MR and uncertainty estimate (Table 5-4). During testing, the GP(2) model performed better than the GP(1) model in terms of RMSE, MARE, and MR. However, compared to the GP(1) model, the GP(2) model resulted in a higher value of uncertainty estimate (Table 5-4). This indicates that increasing the number of mathematical operators, which can be used to define the predictand-predictor relationship, would lead to a better fit (less error) of the function, but at an increasing level of uncertainty instigated by the complex relationships that may exist between the parameters of the model.

In general during training, the NN(BR) model performed better than both the GP(1) and GP(2) models in terms of RMSE, MARE, MR and uncertainty estimates (Table 5-4). While both the GP models were slightly over-predicting (positive MR) K_s, the NN(BR) model was slightly under-predicting (negative MR) K_s (Table 5-4). Because of the very small MR values, both the NN(BR) and the GP models can be considered unbiased. However, during testing, all the models resulted in a negative MR, which indicates that the models are under-predicting the K_s values. Nevertheless, the least RMSE and MARE values of 0.89 and 1.98 were achieved by the GP(2) model. Overall during testing, the GP(1) model resulted in the least uncertainty estimate of 0.26, and the NN(BR) model resulted in the least MR estimate of -0.09. One of the main differences between the NN(BR) model and the GP models adopted in this study is that, in the case of the NN(BR) model, we predefined the initial structure of the network (network inputs, number of hidden layers and hidden neurons). Hence, the structure of the neural networks remains static for each of the resampled datasets, with the values of connections weights of the network optimised every time to fit the corresponding resampled dataset. On the other hand, the GP models are self-organizing in nature, i.e. the model structure and its parameters are evolved simultaneously during the estimation process, learning from the features of the dataset. Therefore, each of the resampled dataset would result in a GP model with different structure and parameters, which would near optimally, fit the corresponding resampled dataset. Figure 5-4 shows the correlation plots between the measured and estimated K_s values by different models during training and testing. Although the regressions (Figure 5-4) are not substantial and account for only a moderate

proportion of the variance, the results are encouraging considering the presence of large scatter between the independent and the dependent variables, and typical in the applications of PTFs (Pachepsky and Rawls, 1999).

One of the important issues in the development of neural network models is the determination of the optimal configuration of the model. The optimal number of hidden neurons and the most significant inputs are usually determined by trial and error method. However, this innate problem in NN modeling can be overcome by the ability of GP in evolving its own model structure with relevant inputs. Table 5-5 shows the most relevant inputs identified by the GP models for the 50 realizations of the training dataset. The values in Table 5-5 represent the percentage of occurrence of each of the variables of the terminal set, in the 50 optimum models. As in the case of GP(1) model, when only additive operators like '+' and '-' are included as part of the functional set, the corresponding percentage of occurrence of sand, silt, clay, and BD, among the optimal models identified for each bootstrapped datasets are 20.6%, 20.6%, 26.5%, and 32.4%, respectively. However, when multiplicative operators, '*' and '/' are added as part of the functional set, as in the case of GP(2) model, the percentage of occurrence of sand content, silt content, and BD increased to 21.7%, 21.4%, and 38.5%, respectively. Nevertheless, the percentage of occurrence of clay content decreased to 18.4%. It can be observed that, both GP(1) and GP(2) identified BD as the most significant input variable (Table 5-5), followed by clay, silt, and sand content in the case of GP(1) model, and by sand, silt, and clay content in the case of GP(2) model. These results indicate that the most relevant input variables (model structure) for estimating K_s using GP is not unique,

rather depends on the kind of mathematical operators (model complexity) that are considered part of the functional set of the GP paradigm to find the predictand-predictor relationship.

5.5 Discussion

Although neural networks have been successfully adopted in developing PTFs for estimating different hydraulic characteristic of soils, their interpretation is often difficult. In neural networks-based PTFs, the knowledge of the predictand-predictor relationships is represented in the form of a weight matrix, which is difficult to comprehend. However, for the same problem, a GP-based PTF gives an explicit equation which can be elucidated with relative ease, depending upon the complexity of the evolved equations. As described in the previous section, in this study, the GP models are combined with the parametric bootstrap method, which generated 50 realizations of the training dataset. Hence, for the 50 realizations of the training dataset, GP-based on self-organizing learning would have arrived at 50 different PTFs. Therefore, for each of the GP models, it is not feasible to present all the 50 different PTFs. Nevertheless, in order to enunciate the transparency of the GP models, both GP(1) and GP(2) models were applied to the actual training data to arrive at corresponding representative PTFs. The GP parameters used in the previous simulation runs (Table 5-3) were retained for this analysis. Equations (5.5) and (5.6) show the representative PTFs obtained by the GP(1) and the GP(2) models, respectively based on the original training dataset. Also, it should be noted that the relationships shown in Equations (5.5) and (5.6) are based on normalized values of Sand, Silt, Clay, BD, and $\log_{10}(K_s)$. From the equations, it can be

observed that the structure of PTFs found by the GP(1) and GP(2) models were different as they are influenced by the kind of mathematical operators used as part of the functional set. Applying Equation (5.5) on the actual training dataset (without resampling), resulted in an RMSE of 1.39; MARE of 0.79; and MR of -0.07. The corresponding values for the testing dataset are 0.94, 2.4, and -0.31, respectively. Similarly Equation (5.6) on the actual training dataset resulted in an RMSE of 1.34; MARE of 0.84; and MR of 0.09. The corresponding values during testing are 0.84, 1.86, and 0.02, respectively.

$$K_s = 1.36 - BD - clay \tag{5.5}$$

$$K_s = (sand \times 0.59) + (1 - BD) \times silt \tag{5.6}$$

Uncertainty in the PTF can result from data, model parameters, and model structure. Data uncertainty stems from natural uncertainty and variability. For calibrated models, the model parameter uncertainty also incorporates the effects of data uncertainties because of the curve-fitting property of the calibration process. Model-structure uncertainty arises from the inability to truly represent the predictand-predictor relationship. These aspects of uncertainty have lead to the concept of equifinality (Beven and Freer, 2001), which argues that there are many different model structures and many different parameter sets within a chosen model structure that may be behavioural or acceptable in reproducing the observed behaviour of a complex environmental system. In most of the PTFs literature, NNs-based PTFs usually account for the model parameter uncertainty by including parametric bootstrap method. In this case, the configuration of

the NN models remains the same for each of the bootstrapped dataset, with the connection weights and bias alone varying depending on the samples present in each of the resampled dataset. However, in this paper, both the model parameter uncertainty and model structure uncertainty are accounted for by combining the GP and parametric bootstrap method. In this case, for each of the bootstrapped dataset, both the model structure and the model parameters are optimized in unison by GP. Hence, the value of uncertainty of the NN(BR) model (Table 5-4), indicates the model parameter uncertainty only, while the GP models uncertainty (Table 5-4) indicates both the model parameter and model structure uncertainty.

For the GP models, the relative contribution of the model parameter uncertainty and the model structure uncertainty to the total uncertainty, reported in Table 5-4 is also examined in this study. In order to accomplish this, instead of simultaneously evolving both the model structure and the model parameters using the self-organizing nature of GP, by pre-defining the model structure and optimizing its corresponding model parameters alone, the uncertainty due to model parameters can be determined. By comparing this uncertainty with the total uncertainty (model parameter and model structure) reported in Table 5-4, it is possible to determine the relative contribution of the model parameter uncertainty and model structure uncertainty to the total uncertainty. The model structures based on Equations (5.5) and (5.6), which represent the PTFs obtained by the GP(1) and the GP(2) models, respectively for the original training dataset, were chosen to be the representative model structures for all the 50 resampled datasets. Keeping the model structure static, their respective model parameters alone were

optimized for each of the 50 resampled datasets. Equation (5.5) and Equation (5.6) have three and four parameters respectively, which need to be optimized. The uncertainty estimate and the performance statistics were calculated as outlined earlier. When Equation (5.5) (based on GP(1)) was used as the representative model structure, during training, it resulted in an uncertainty of 0.11, RMSE of 0.85, MARE of 0.78, and MR of 0. The corresponding values during testing were 0.09, 0.93, 2.31, and -0.26, respectively. Comparing these statistics with that of the GP(1) model statistics (Table 5-4), it can be concluded in general that, keeping the model structure static and optimizing the model parameters alone, both during training and testing resulted in less uncertainty but at the expense of higher error statistics. The uncertainty estimates of 0.11 during training, and 0.09 during testing are relatively small when compared to that of the uncertainty estimates of the GP(1) model (Table 5-4), which resulted in uncertainty estimates of 0.27 and 0.26, during training and testing, respectively. When Equation (5.6) (based on GP(2)) is used as the representative model structure, during training, it resulted in an uncertainty of 0.09, RMSE of 0.77, MARE of 0.76, and MR of -0.01. The corresponding values during testing were 0.10, 0.87, 1.89, and -0.01, respectively. Comparing these statistics with that of the GP(2) model statistics (Table 5-4), keeping the model structure static and optimizing the model parameters alone, during training resulted in less uncertainty but with higher error statistics. Nevertheless during testing, compared to the GP(2) model, the performance of the model with static model structure and optimized model parameters, resulted in considerably less uncertainty estimate and also relatively better error statistics. Hence in general, it can be stated that, compared to the models with optimized model structure and model parameters (GP(1) and GP(2)), the models with

static model structure and optimized model parameters, resulted in markedly lower uncertainty estimates. As argued earlier, since the former models accounts for both the model parameter and model structure uncertainty, and the latter models accounts for the model parameter uncertainty alone, it can be concluded that, compared to the model parameter uncertainty, the contribution of model structure uncertainty to the actual uncertainty is more significant. Based on the above analysis, it can be stated that for ensemble modeling of K_s using GP, for each of the resampled datasets, the choice between (i) keeping the model structure static and optimizing the model parameters alone, and (ii) self-organizing both the model structure and model parameters, should be made considering the kind of uncertainty (model parameter and/or model structure) that needs to be accounted for in the ensemble modeling of K_s.

5.6 Summary and Conclusions

In this study, the utility of GP as a model induction engine for deriving PTFs for estimating K_s as a function of sand, silt, clay contents, and bulk density is explored. Out of the 314 samples derived from the UNsaturated SOil hydraulic DAtabase (UNSODA) database, 199 samples are used for the calibration purpose and the remaining 114 samples are used for validating the developed models. Two different GP models, GP(1) and GP(2), with different combination of mathematical operators in the functional set are developed. The GP(1) model uses only additive (+,-) operators as part of the functional set, and the GP(2) model uses both additive (+,-) and multiplicative (*, /) operators as part of the functional set. This exercise was carried out to evaluate the performance of the GP models with varying levels of mathematical operators (complexity) that can be used

to define the predictand-predictor relationship. The performance of the GP(1) and GP(2) models are compared with a neural networks model employing Bayesian-regularization (BR) algorithm for training the networks. The BR algorithm has better generalization property as it minimizes both the mean sum of squares of the network errors and the mean of the sum of squares of the network weights and bias.

The performances of the models are evaluated in terms of root mean squared error (RMSE), mean absolute relative error (MARE), mean residual (MR), and model uncertainty. The uncertainty of the models is evaluated by combining the models with the non-parametric bootstrap method. Fifty different bootstrapped datasets are created by statistical resampling of the training dataset, and the NN and GP models are applied on these bootstrapped datasets, from which the average error estimates and uncertainty of the model predictions are evaluated. Results from the study indicate that the GP(2) model resulted in the least MRE and MR estimates, signifying the less bias attached to the model. The relatively better performance of the GP models, compared to the NNs model, may be attributed to the self-organizing nature of the model, in which both the model structure and parameters are evolved simultaneously during the estimation process, learning the features of the dataset. Increasing the number of mathematical operators in the functional set of the GP models has been found to lead to a better fit of the function, but at an increasing level of uncertainty. This indicates, if not unfeasible, it is difficult to achieve both higher prediction accuracy and less uncertainty in tandem. The results presented in this study, in general indicate that the GP is a promising tool for developing PTFs for estimating K_s.

Analyzing the optimal equations identified by the GP models for each of the bootstrapped dataset, BD is the most significant input in characterizing the K_s for both the GP(1) and GP(2) models. In the case of GP(1) model, in the order of importance, BD is followed by clay, sand, and silt contents. However for the GP(2) model, in the order of importance, BD is followed by sand, silt, and clay contents. These results indicate that the most relevant input variables for estimating K_s is not unique, rather depends on the mathematical operators that are used as part of the functional set of the GP paradigm. Also, it has been shown that the uncertainty reported by the NN(BR) model is only the model parameter uncertainty, whereas the uncertainty reported by the GP models include both the model parameter and model structure uncertainty. Examining the relative contribution of model structure uncertainty and model parameter uncertainty to the total uncertainty estimated by the GP models, it is been shown that, compared to the model parameter uncertainty, the uncertainty due to the model structure dominates the total uncertainty of the GP models. The study reported in this paper is a first step to evaluate the utility of GP in developing PTFs. The results of the study need to be further explored by extending the GP models to different datasets with different functional and terminal sets. In this regard, analyzing the performance of grammar-guided GP in developing PTFs would be of particular interest.

5.7 Acknowledgements

Fundings for this project were provided by the National Science and Engineering Research Council of Canada (NSERC) to AE and BCS. The authors thank

the Associate Editor (Yakov A. Pachepsky) and three anonymous reviewers for their insightful reviews.

5.8 References

Babovic, V., and Keijzer, M. (2000). "Genetic programming as model induction engine." *J. Hydroinformatics,* 2, 35-60.

Babovic, V., and Abbott, M. B. (1997). "Evolution of equation from hydraulic data: Part I-Theory." *J. Hydraul. Res.,* 35, 1-14.

Beven, K., and Freer, J. (2001). "Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology." *J. Hydrol.,* 249, 11-29.

Bouma, J. (1989). "Using soil survey data for quantitative land evaluation." *Adv. Soil Sci.,* 9, 177-213.

Cosby, B.J., Hornberger, G. M., Clapp, R. B., and Ginn, T. R. (1984). "A statistical exploration of the relationships of soil moisture characteristics to the physical properties of soils." *Water Resour. Res.,* 20, 682-690.

Coulibaly, P. (2004). "Downscaling daily extreme temperatures with genetic programming." *Geophys. Res. Lett.,* 31, L16203, doi:10.1029/2004GL020075.

Demuth, H., and Beale, M. (2001). *Neural network toolbox learning. For use with MATLAB*. The Math Works Inc, Natick, Mass.

Efron, B., and Tibshirani, T. J. (1993). *An introduction to bootstrap*. Chapman and Hall, New York, NY.

Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). *Artificial intelligence through simulated evolution*. Ginn, Needham Height.

Folino, G., Pizzuti, C. and Spezzano, G. (2006). "GP ensembles for large-scale data classification." *IEEE Trans. Evol. Comp.,* 10, 604-616.

Gupta, S. C., and Larson, W. E. (1979). "Estimating soil water characteristic from particle size distribution, organic matter percent, and bulk density." *Water Resour. Res.,* 15, 1633-1635.

Haykin, S. (1999). *Neural networks: A comprehensive foundation*, 2nd ed. MacMillan, New York.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. University of Michigan, Illinois.

Hong, Y. S., and Rosen, M. R. (2002). "Identification of an urban fractured-rock aquifer dynamics using an evolutionary self-organizing modelling." *J. Hydrol.,* 259, 89-104.

Hong, Y. S., White, P. A., and Scott, D. M. (2005). "Automatic rainfall recharge model induction by evolutionary computational intelligence." *Water Resour. Res.,* 41, W08422, doi:10.1029/2004WR003577.

Iba, H. (1999). "Bagging, boosting, and bloating in genetic programming." In W. Banzhaf et al. (Eds.) *Proceedings of the Genetic and Evolutionary Computation Conference*, Orlando, Florida, USA, 13-17 July 1999, Morgan Kaufmann, 2:1069-1076.

Keijzer, M., and Babovic, V. (2000). "Genetic programming, ensemble methods and the bias/variance tradeoff – Introductory investigations." p. 76-90. In R. Poli et al. (ed.)

Proc. Of EuroGP'2000. Vol. 1802. 15-16 Apr. 2000. Edinburgh. Springer-Verlag, Berlin.

Khu, S. T., Liong, S. Y., Babovic, V., Madsen, H., and Muttil, N. (2001). "Genetic programming and its application in real-time runoff forecasting." *J. Am. Water Resour. Assoc.,* 8, 201-220.

Koza, J. R. (1992).*Genetic programming: On the programming of computers by means of natural selection.* The MIT Press, Cambridge, Massachusetts.

Klute, A. (1986). "Methods of soil analysis, Part 1." *Physical and mineralogical methods*, Agron. Monogr. No. 9, ASA and SSSA, Madison, WI.

Leij, F., Schaap, M. G., and Arya, L. M. (2002). "Water retention and storage: Indirect methods." p. 1009-1045. *In* J.H. Dane and G.C. Topp (ed.) *Methods of soil analysis: Part 4-Physical methods*. SSSA Book Ser. No. 5. SSSA, Madison, WI.

Leij, F. J., Alves, W. J., van Genuchten, M. Th., and Williams, J. R. (1996). "Unsaturated Soil Hydraulic Database, UNSODA 1.0 User's Manual." *Report EPA/600/R-96/095. US Environmental Protection Agency*, Ada, Oklahoma, 103 pp.

Makkeasorn, A., Chang, N. B., Beaman, M., Wyatt, C., and Slater, C. (2006). "Soil moisture estimation in semiarid watershed using RADARSAT-1 satellite imagery and genetic programming." *Water Resour. Res.,* 42, W09401, doi:10.1029/2005WR004033.

McBratney, A. B., Minasny, B., Cattle, S. R., and Vervoort, R. W. (2002). "From pedotransfer functions to soil inference systems." *Geoderma,* 109, 41-73.

McKenzie, N. J., and Jacquier, D. W. (1997). "Improving the field estimation of saturated hydraulic conductivity in soil survey." *Aust. J. Soil Res.,* 35, 803-825.

Minasny, B., McBratney, A. B., and Bristow, K. L. (1999). "Comparison of different approaches to the development of pedotransfer functions for water retention curves." *Geoderma,* 93, 225-253.

Nemes, A., Rawls, W. J., and Pachepsky, Ya. (2005). "Influence of organic matter on the estimation of saturated hydraulic conductivity." *Soil Sci. Soc. Am. J.,* 69, 1330-1337.

Pachepsky, Ya., and Rawls, W. J. (1999). "Accuracy and reliability of pedotransfer functions as affected by grouping soils." *Soil Sci. Soc. Am. J.,* 63, 1748-1757.

Pachepsky, Ya., Rawls, W. J., Gimenez, D., and Watt. J. P. C. (1998). "Use of soil penetration resistance and group method of data handling to improve soil water retention estimates." *Soil and Till. Res.,* 49, 117-126.

Pachepsky, Ya., Timlin, D. J., and Varallyay, G. (1996). "Artificial neural networks to estimate soil water retention from easily measurable data." *Soil Sci. Soc. Am. J.,* 60, 727-773.

Parasuraman, K., Elshorbagy, A., and Carey, S. K. (2007). "Modeling the dynamics of evapotranspiration process using genetic programming." *Hydrol. Sci. J.,* 52, 563-578.

Poli, R. (2003). "A simple but theoretically-motivated method to control bloat in genetic programming." In C. Ryan et al. (Eds.) *Proceedings of EuroGP 2003*, Springer, Berlin, 204-217.

Rawls, W. J., Gish, T. J., and Brakensiek, D. L. (1991). "Estimating soil water retention from soil physical properties and characteristics." *Adv. Soil Sci.,* 9, 213-234.

Rawls, W. J., and Brakensiek, D. L. (1983). "A procedure to predict Green and Ampt infiltration parameters." p. 102-112. In *Adv. in Infiltration*. ASAE, St. Joseph, MI.

Savic, D. A., Walters, G. A., and Davidson, J. W. (1999). "Genetic programming approach to rainfall-runoff modelling." *Water Resour. Mgmt.,* 13, 219-231.

Saxton, K. E., Rawls, W. J., Romberger, J. S., and Papendick, R. I. (1986). "Estimating generalized soil-water characteristics from texture." *Soil Sci. Soc. Am. J.,* 50, 1031-1036.

Schaap, M. G., and Bouten, W. (1996). "Modeling water retention curves of sandy soils using neural networks." *Water Resour. Res.,* 32, 3033-3040.

Schaap, M. G., and Leij, F. J. (1998). "Database-related accuracy and uncertainty of pedotransfer functions." *Soil Sci.,* 163, 765-779.

Schaap, M. G., Leij, F. J., and van Genuchten, M. Th. (2001). "ROSETTA: a computer program for estimating soil hydraulic parameters with hierarchical pedotransfer functions." *Soil Sci.,* 163, 765-779.

Schwefel, H. P. (1981). *Numerical optimization of computer models*. Wiley, Chichester.

Silva, S. (2005). GPLAB – a genetic programming toolbox for MATLAB, http://gplab.sourceforge.net

Silva, S., and Costa, E. (2004). "Dynamic limits for bloat control." In K. Deb et al. (Eds.) *Proceedings of GECCO 2004*, Springer, Berlin, 666-677.

Soule, T., and Foster, J. A. (1999). "Effects of code growth and parsimony pressure on populations in genetic programming." *Evol. Comp.,* 6, 293-309.

Tamarai, S., Wösten, J. H. M., and Ruiz-Suarez, J. C. (1998). "Testing an artificial neural network for predicting soil hydraulic conductivity." *Soil Sci. Soc. Am. J.,* 60, 1732-1741.

Ungaro, F., Calzolari, C., and Busoni, E. (2005). "Development of pedotransfer functions using a group method of data handling for the sol of the Pianura Padano-Veneta region of North Italy: water retention properties." *Geoderma,* 124, 293-317.

Van Genuchten, M. Th., Leij, F. J., and Lund, L. J. (1992). "On estimating the hydraulic properties of unsaturated soils." P.1-14. In M.Th. van Genuchten et al. (ed.) *Indirect methods for estimating the hydraulic properties of unsaturated soils*. Proc. Int. Workshop. Riverside, CA. 11-13 Oct. 1989. Univ. of California, Riverside.

Vereecken, H., Maes, J., and Feyen, J. (1990). "Estimating unsaturated hydraulic conductivity from easily measured soil properties." *Soil Sci.,* 149, 1-12.

Whigham, P. A., and Craper, P. F. (2001). "Modelling rainfall-runoff using genetic programming." *Math. Comput. Modell.,* 33, 707-721.

Wösten, J. H. M. (1990). Use of soil survey data to improve simulation of water movement in soils. Ph. D. thesis. Univ. of Wageningen, Netherlands.

Wösten, J. H. M., Pachepsky, Ya., and Rawls, W. J. (2001). "Pedotransfer functions: Bridging gap between available basic soil data and missing soil hydraulic characteristics." *J. Hydrol.,* 251, 123-150.

Table 5-1 Descriptive statistics and correlation matrix of the training dataset

	Sand (%)	Silt (%)	Clay (%)	BD (gm cm^{-3})	$log_{10}K_s$
Minimum	1.80	0.20	0.00	0.59	-1.19
Maximum	99.10	81.40	63.00	1.97	4.44
Median	52.80	25.55	14.95	1.52	1.94
Mean	54.33	27.96	17.71	1.47	1.87
SD	30.53	20.73	15.46	0.25	1.08
CV	0.56	0.74	0.87	0.17	0.58
Correlations					
Sand (%)	1.00				
Silt (%)	-0.89	1.00			
Clay (%)	-0.79	0.41	1.00		
BD (gm cm^{-3})	0.47	-0.38	-0.42	1.00	
$log_{10}K_s$	0.52	-0.39	-0.50	-0.11	1.00

Table 5-2 Descriptive statistics and correlation matrix of the testing dataset

	Sand (%)	Silt (%)	Clay (%)	BD (gm cm^{-3})	$log_{10}K_s$
Minimum	0.10	0.30	0.10	0.49	-0.84
Maximum	99.60	80.70	54.40	1.76	3.58
Median	54.75	27.30	11.25	1.49	1.91
Mean	51.77	34.58	13.65	1.48	1.78
SD	33.21	26.83	11.74	0.16	0.89
CV	0.64	0.78	0.86	0.11	0.50
Correlations					
Sand (%)	1.00				
Silt (%)	-0.95	1.00			
Clay (%)	-0.67	0.39	1.00		
BD (gm cm^{-3})	0.34	-0.24	-0.41	1.00	
$log_{10}K_s$	0.39	-0.40	-0.20	-0.07	1.00

Table 5-3 GP Parameters

GP Parameter	Value
Population Size	20
Initialization Method	Ramped half-and-half
Sampling Method	Roulette
Maximum Initial Tree Depth	8
Probability of Crossover, P_c	0.6
Probability of Mutation, P_m	0.3
Cost Function	RMSE
Number of Generations	1000

Table 5-4 Performance statistics of different models in estimating Ks

Model	Training				Testing			
	Uncertainty	RMSE	MARE	MR	Uncertainty	RMSE	MARE	MR
NN(BR)	0.26	0.61	0.55	-0.01	0.27	1.04	2.23	-0.09
GP(1)	0.27	0.83	0.76	0.02	0.26	0.90	2.24	-0.22
GP(2)	0.27	0.70	0.68	0.02	0.30	0.89	1.98	-0.13

Table 5-5 Percentage of different input variable selection in the GP models

	Sand	Silt	Clay	BD
GP(1)	20.6	20.6	26.5	32.4
GP(2)	21.7	21.4	18.4	38.5

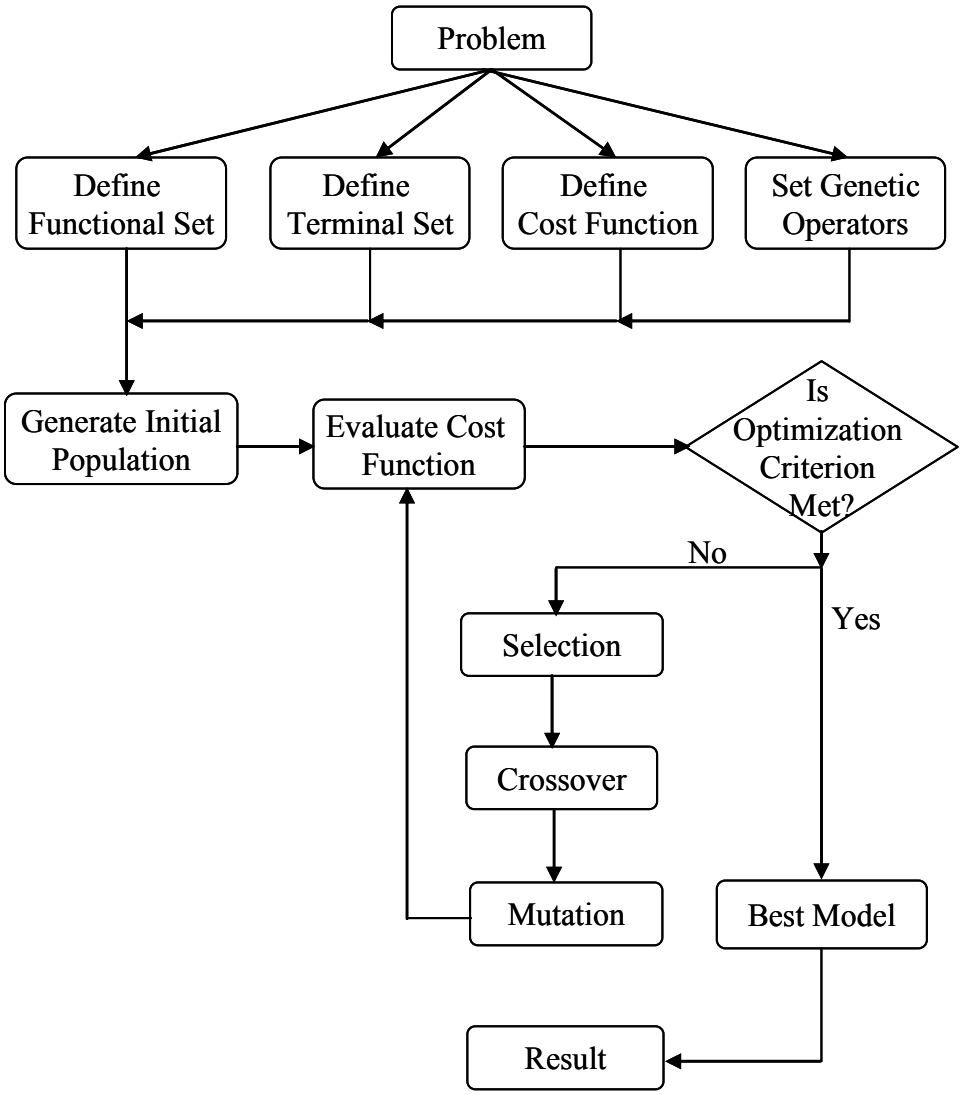

Figure 5-1 Flowchart of the GSR paradigm

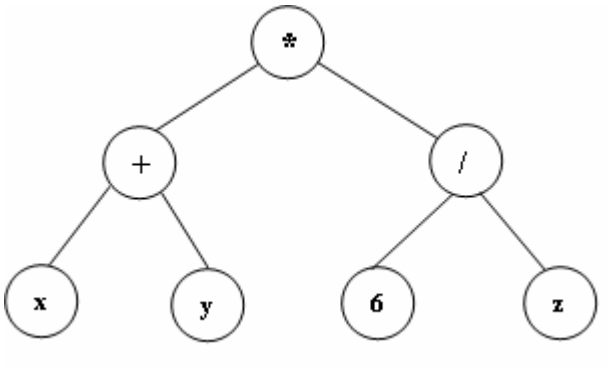

Figure 5-2 Sparse tree notation

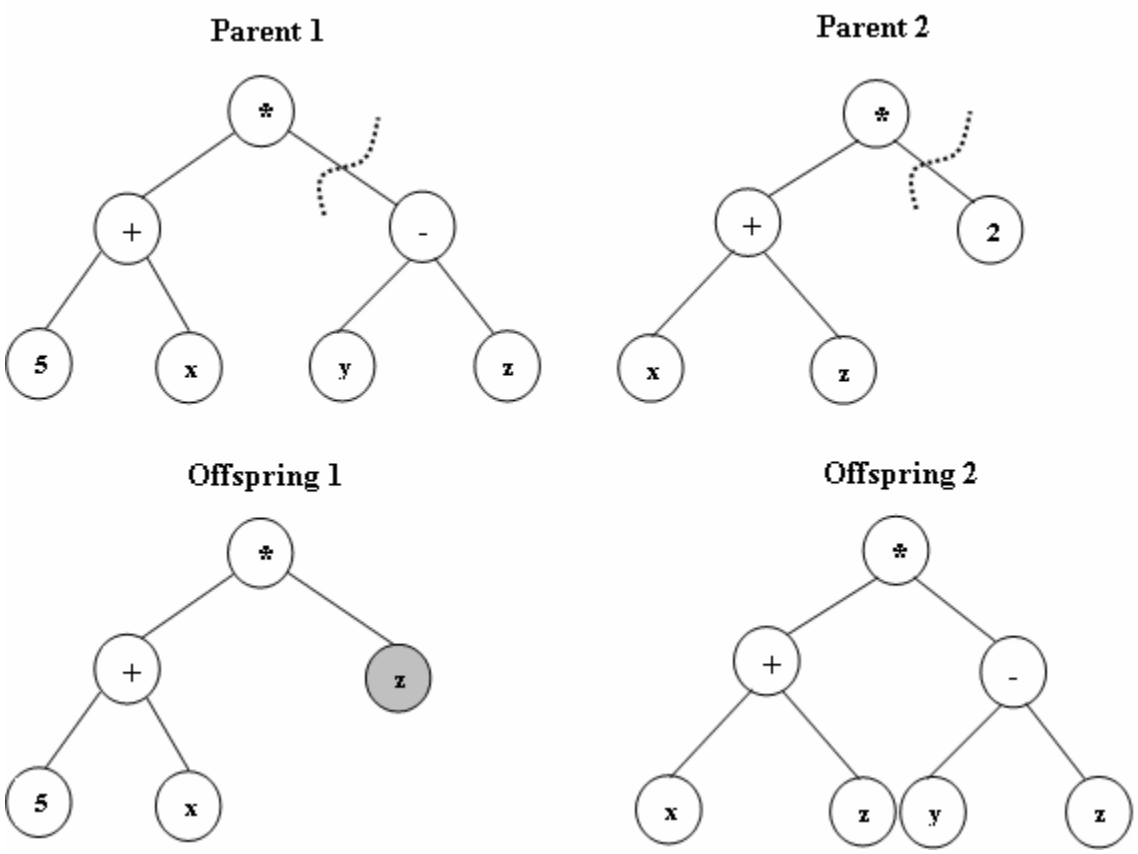

Figure 5-3 Crossover coupled with mutation. The dashed line indicates the Crossover point and the shaded region represents the mutated node.

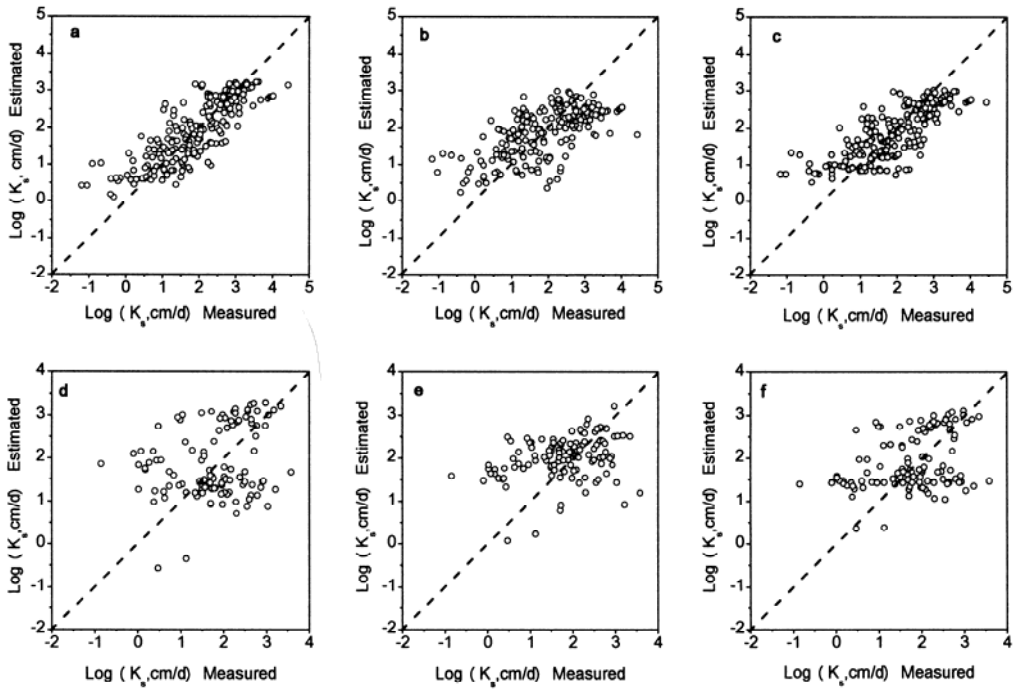

Figure 5-4 Comparison of measured and estimated K_s by different models during training [(a) NN(BR), (b) GP(1), and (c) GP(2)]; and testing [(d) NN(BR), (e) GP(1), and (f) GP(2)]

Chapter 6 Conclusions

6.1 Summary of the Thesis

In general, this thesis comprises of two parts; the first part, which covers chapter 2 and chapter 3, is motivated by the idea of extending the usefulness of data-driven models beyond forecast applications as a tool for scientific investigations. Chapters 2 and 3 identify some of the possible methods and tools that can be adopted to bring transparency to the way by which the data-driven models arrive at a solution. The second part, which covers chapters 4 and 5, is motivated by the idea of improving the reliability of the data-driven models by identifying ways for incorporating uncertainty estimates as part of the data-driven model building exercise.

In chapter 2, a novel artificial neural network (ANN) model, named "spiking modular neural networks (SMNNs)" has been proposed. The SMNNs are based on the concepts of both self-organizing networks and modular networks. Two variants of SMNNs, employing (i) competitive learning, and (ii) self-organizing maps, have been developed and tested. Contrary to the traditional neural network models, that does not consider the presence of discontinuity in the input-output mapping space, the modular nature of the proposed SMNNs is shown to account for this discontinuity by developing domain dependent input-output relationships. The performances of the SMNNs are evaluated on two distinctly different case-studies, namely, (i) streamflow modeling, and (ii) actual evapotranspiration modeling. The performance of the SMNNs is compared to that of the regular feed forward neural network (FFNN) model as it is the most widely

adopted neural network model in water resources applications. For both case-studies, the SMNNs are shown to perform better than the conventional FFNN model. Also, the SMNNs are shown to be effective in discretizing the complex mapping space into simpler domains that can be learnt with relative ease. The study demonstrated how the usefulness of SMNNs can be extended beyond forecast applications as a tool for scientific investigation by demonstrating the way SMNNs, as a data-driven model, was able to reiterate the fact that different combination of state variables can satisfy the energy balance equation. The study reported in chapter 2 is a step in the direction towards developing multiple local models rather than a single global model for hydrological processes.

In chapter 3, the ability of another promising data-driven modeling technique, namely genetic programming (GP), has been evaluated with regards to its ability to promote transparency in data-driven models. This study was founded on the hypothesis that the robustness of GP to evolve its own model structure with relevant parameters could aid in understanding and improving our knowledge of the predictand-predictor relationship. The hypothesis was tested by applying GP to model the dynamics of the actual evapotranspiration process from two case-studies with different topographic conditions. The performances of the GP models were compared with ANN models and the traditional Penman-Monteith (PM) method. Results from the study indicated that both data-driven models, GP and ANNs, performed better than the PM method. However, the performance of the GP model is comparable with the ANNs model. The ability of GP to arrive at an explicit model structure for modeling the site-specific actual

evapotranspiration process has been demonstrated in this study. From the insights gained by analyzing the GP-evolved equations, it was found that the GP-evolved equations are dominated by net-radiation (NR) and ground temperature (GT), indicating that NR and GT are the most important state variables for characterizing the evapotranspiration process. This is consistent with the findings from the previous study (chapter 2); where NR and GT alone were shown to explain most of the variance in latent heat (LE) flux using SMNNs. It is argued in this thesis that NR is the driving variable during energy-limited conditions, and GT, as a surrogate for soil moisture, is the driving variable during supply-limited conditions. The rationale for this argument is based on the strong link between the soil thermal properties and moisture status.

Chapter 4 has demonstrated the improvement in the reliability that can be achieved by adopting a field scale model as against a global scale model. For estimating the saturated hydraulic conductivity at two distinct sites, field scale models were developed using neural network ensembles. Two variants of the field scale models, one employing bagging algorithm, and the other employing boosting algorithm, were developed and tested with the objective of identifying the relative merits and demerits of adopting these resampling algorithms for constructing neural network ensembles. The performance of the field scale models were compared with a published global neural network model, ROSETTA. For the field scale models, compared to the model employing conventional bagging algorithm, the model employing boosting algorithm has been shown to produce networks with less bias. In general, the local-scale models have been shown to be more reliable than the global-scale models. Bearing in mind the

presence of several large (global) scale models, the findings from this study reiterate that such models may be best suited for providing trends at global scale, but may be of little use for more practical applications (salt balance and water balance) at field-scales.

It has been widely acknowledged in the literature that the uncertainty in geophysical modeling can manifest in terms of natural randomness, data, model parameters, and model structure. In chapter 5, a methodology for improving the reliability of geophysical models by accounting for the influence of model-structure uncertainty has been proposed. In this case, contrary to the traditional approaches, both the model structure and its parameters were assumed to be imperfectly known, and self-organizing algorithms were used to search from a pool of model structures and model parameters to arrive at an ensemble of possible combinations of model structure and parameters, from which the actual uncertainty was calculated. The proposed methodology was evaluated in developing pedotransfer functions for estimating the saturated hydraulic conductivity of soils. A dataset from the UNsaturated SOil hydraulic DAtabase (UNSODA) has been considered in this study. ROSETTA, a neural network based pedotransfer function, was used for comparison purposes because of their previously established utility in geophysical literature. The uncertainty due to model structure has been shown to be larger than the uncertainty due to model parameters. Also, it has been demonstrated that increasing the model complexity may lead to a better fit of the function, but at the cost of an increasing level of uncertainty.

In summary, this thesis identifies some of the possible methods and tools that can be adopted to accomplish Tasks 3 and 5, outlined in chapter 1. The methods and tools proposed in this thesis are not claimed to be exhaustive, and hence may not in depth address all the pertinent issues with regard to the above tasks. Nevertheless, this study would serve as a catalyst for future studies in this direction, by exploring multiple avenues for accomplishing the above tasks. The applications that were adopted in this thesis to test the validity and the utility of the proposed tools and methods were not restricted to the oil sands reclamation areas. Other relevant case studies were also considered with the purpose of strengthening the presentations, as the overriding objective of this thesis is to explore different methods and tools that can be adopted to: (i) extend the usefulness of the system-theoretic models beyond forecast applications as a tool for scientific investigation; and (ii) improve the reliability of the system-theoretic models by identifying ways for incorporating uncertainty estimates as part of the data-driven model building exercise. Nevertheless, it is expected that the methods and tools identified in this thesis can be extended to characterize diversified geophysical processes pertaining to the oil sands reclamation.

6.2 Research Contribution

The contribution of this thesis to the field of hydrology can be categorized under two levels of contribution scale, not regarding their significance but rather their conceptual level: level-1 contribution and level-2 contribution.

6.2.1 Level-1 contribution

At the conceptual level, the first major contribution of this thesis is the spiking modular neural networks (SMNNs) proposed in chapter 2. The SMNNs is shown to be conceptually different from the traditional neural network models. The SMNNs have the ability to break down a complex mapping space into simpler domains that can be learnt with relative ease. The SMNNs develop domain dependent input-output relationships to account for the discontinuity in the input-output mapping space. In this thesis, it was also shown that, for a given input space, topology learning may not be of significant help in improving the performance of the modular neural networks. The thesis also highlighted how the SMNNs can be used beyond forecast applications as a tool for scientific investigations by identifying the patterns in the mapping space. The SMNNs proposed in this thesis can be considered a step in the direction to develop multiple local models rather than a single global model for geophysical processes.

The second major contribution of this thesis is the methodology, proposed in chapter 5, to evaluate the effect of model structure uncertainty in geophysical modeling. Although model structure uncertainty is acknowledged to be an important factor in geophysical modeling, the traditional approach to geophysical model uncertainty has been to hypothesize a deterministic model structure and treat its parameters as being imperfectly known. The uncertainty estimated by these traditional approaches is just a small portion of the actual uncertainty; they neglect the uncertainty associated with model structure by assuming it to be deterministic. This thesis offers one possible solution to the above problem by developing a framework based on self-organizing algorithms and

statistical resampling technique to account for the model structure uncertainty in geophysical modeling. In this thesis, it was established that the uncertainty due to model structure is larger than the uncertainty due to model parameters, and an increase in the model complexity is shown to increase the predictive ability of the model, but at an increasing level of uncertainty.

6.2.2 Level-2 contribution

This level of contribution also adds to the field of hydrology at the conceptual level with less generality than the level-1 contribution. The following are the Level-2 contribution of this thesis: (1) this thesis highlighted that net-radiation and ground temperature are the most important state variables for characterizing the eddy covariance-measured evapotranspiration flux; (2) in this thesis it is argued, and subsequently validated using literature, that ground temperature can be considered as a surrogate variable of soil moisture due to the strong link between the soil thermal properties and moisture status; (3) underscored the utility of genetic programming to promote transparency in data-driven hydrological modeling, and thereby improving our knowledge of the predictand-predictor relationship; and (4) highlighted the advantages of adopting local models compared to global models for characterizing geophysical processes (chapter 2) and properties (chapter 4).

6.3 Possible Research Extension

Improvements in the approaches, methodologies, and models developed in this thesis are possible at various levels and in different parts of this research. Some of the opportunities for possible future research directions are briefed below:

In the modular neural networks proposed in this thesis, other methods and techniques from pattern recognition such as the fuzzy c- means clustering can be investigated to enhance the clustering process. Also, the use of global optimization methods like genetic algorithms can be evaluated to optimize the network weights and bias. In this thesis, the optimal number of clusters and network parameters are identified by the trail-and-error. It might be of interest in future studies to investigate alternative methods to identify the above parameters more objectively. Also, more research on providing physical interpretation to the patterns identified by the SMNNs is warranted in future studies. Evaluating the effect of modularization on the predictive uncertainty of the model would also be of interest.

In this thesis, conversion of Penman-Monteith estimates of potential evapotranspiration to their actual evapotranspiration counterparts was not attempted due to the inherent limitations and uncertainty (outlined in chapters 2 and 3) associated with such conversions. It might be worth investigating how the actual estimates of evapotranspiration estimated considering the soil moisture limitations, would compare with the actual evapotranspiration estimates provided by the system-theoretic models proposed in this study.

In developing the neural network ensembles, this thesis employed simple averaging to combine the outputs from different networks. The relative merits and demerits of adopting alternative method of combining networks, like the weighted averaging and stacking, needs to be explored. Also, more analysis regarding uncertainty propagation through a hydrological system, as a function of the uncertainty in the system-theoretic model predicted values, needs to be ascertained. i.e. evaluating the applicability of the system-theoretic model predicted values in modeling hydrological processes.

The methodology to account for the effect of the model structure uncertainty in geophysical modeling needs to be further expanded to identify the tradeoffs, if any, between models' structural complexity, accuracy, and uncertainty. Also, the usefulness of more versatile self-organizing algorithms like evolutionary polynomial regression, and grammar-guided genetic programming should be sought with regards to their ability in identifying the patterns in the input-output space.

6.4 Study Limitations

Several limitations can be noted with regards to the methods and analysis adopted in this thesis. In this thesis, a unique application with a unique dataset was not benchmarked to evaluate the relative performance of the different methods and tools proposed in this thesis. Nevertheless, this study adopted diverse real-world applications to strengthen the presentation, as the overriding objective of this thesis was not to pit one method against the other, rather to explore different methods and tools that can be

adopted to promote transparency and reliability in data-driven model building exercise. However, the methods and tools proposed in this thesis can be easily extended to evaluate their relative performance on a wide range of geophysical applications.