Developing Artificial Neural Networks for Water Quality Modelling and Prediction

Robert James May

BEng (Chem) Hons, BSc

Thesis submitted to The University of Adelaide
School of Civil, Environmental & Mining
Engineering in fulfilment of the requirements
for the degree of Doctor of Philosophy

Abstract

Modelling water quality within complex, man-made and natural environmental systems can represent a challenge to practitioners. Many conventional modelling tools are not capable of representing the complexities of physical and chemical processes often observed in these systems. Consequently, there has been a great deal of interest in the application of computational intelligence techniques, such as artificial neural networks (ANNs). However, “black-box” approaches, such as ANN modelling, are often criticised due to a perceived lack of transparency in the model development methodology. This research has therefore focussed on improving the tools and techniques that are used in the development of ANN models for water quality prediction and forecasting.

The body of research presented in this thesis is described by several peer reviewed articles. These articles describe the theoretical basis and practical context for the ANN model development techniques that have been proposed and applied as a part of this research. Specifically, the ANN development framework has been further enhanced by this research through the development of novel approaches to perform two key tasks: input variable selection (IVS) and data splitting.

The IVS problem is to select variables as ANN inputs from a number of potential candidates, so as to minimise the number of inputs, but maximise the predictive performance of the model. A forward-selection approach for IVS has been examined that is based on partial mutual information (PMI), which can identify an optimal set of variables to use as inputs to ANN models, given a set of candidate variables. Of particular concern is that the use of MI in place of the more traditionally used correlation, provides a more appropriate basis for the selection of inputs based on non-linear relevance. Moreover, the accuracy of MI estimates for a given sample size is difficult to determine. Quantifying the accuracy of MI estimates is necessary to determine critical values of MI, since this forms the basis for of the termination criterion that stops the forward selection process.

Novel termination criteria were developed that alternatively determine the optimum number of candidate input variables. In comparison to the existing approach, which is based on a computationally expensive, yet potentially inaccurate bootstrap approach, the alternative criteria were found to both reduce the computational requirements and increase selection accuracy of the PMI-based IVS approach, resulting in a much improved algorithm.

Data splitting is an essential part of ANN model development, as the available modelling data must be partitioned into subsets for training, testing and validation. Depending on the data splitting method employed, the data split can have a significant effect on model performance, or reduce confidence in performance assessment. A popular method based on clustering of the self-organizing map (SOM) was examined. The approach was found to be sensitive to SOM size and the manner in which samples are drawn from within the SOM units. However, despite an optimal number of partitions, the SOM can generate partitions that are non-uniformly distributed, and which differ in size and shape. Although conventional rules to increase the sampling rate within larger clusters can reduce variance, the remaining variance can still be significant.

A hybrid algorithm called SOMPLEX was developed, which combines clustering on the SOM, and the DUPLEX algorithm used to perform intra-cluster sampling. DUPLEX is a fully deterministic algorithm that generates a representative sample, regardless of the size or distribution of data within a SOM cluster. For several example applications to predicting water quality, SOMPLEX was found to generate representative data for training, testing and validation, with no variation. The hybrid SOMPLEX approach combines the strengths of the two individual data splitting algorithms, in that the clustering on the SOM reduces the operational complexity, and the DUPLEX sampling improves on random sampling of SOM units to reduce sample variability and increase the representativeness of datasets generated.

In terms of the overall ANN development framework, the outcomes of this research have been an increased understanding of how to best implement ANN techniques, and an appreciation for their place within the context of a water quality modelling toolkit, which comprises both conventional and non-conventional modelling approaches. It was also observed that although the ANN modelling paradigm is quite powerful, it is not without limitations. Many of the limitations and problems encountered with ANN model development are more indicative of the application, rather than the modelling approach itself.

Statement of Originality

I, *Robert James May*, hereby declare that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

The author acknowledges that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the Universitys digital research repository, the Library catalogue, the Australasian Digital Theses Program (ADTP) and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signed: .... Date: ....

Acknowledgement

Above all, I wish to express my profound gratitude towards my supervisors, Professor Graeme Dandy and Professor Holger Maier, for their invaluable guidance, support and encouragement throughout the course of this research.

The author is grateful for the financial support provided by the CRC for Water Quality and Treatment, the technical assistance from the research team for Project 2.5.0.1, and the project leadership of Mr Michael Holmes and Dr Christopher Chow. I thank United Water for their support of this research, in particular Dr John Nixon for his contribution as my industry supervisor, and Dr Stephanie Rinck-Pfeiffer, for her patience while I completed this thesis.

Thanks to the all the staff within the School of Civil, Environmental & Mining Engineering, in particular Dr Stephen Carr for his assistance with software development. The author also wishes to acknowledge that a significant part of this research was aided by the facilities provided by the South Australian Partnership for Advanced Computing.

Many thanks to all of my fellow post-graduate students within the School, for the Friday afternoon drinks and carpet bowls—I blame you all for making the experience so enjoyable that I lost all motivation to finish. Thanks especially to Ms. Gayani Fernando, for sharing a mutual love of neural networks; Dr Matthew Gibbs and Mr Darren Broad, for the more than occasional Tuesday drinks; and I am truly indebted to Mr Michael Leonard for a year of Saturdays spent writing up, and many other enjoyable discussions over the years.

Finally, I acknowledge that credit for any of my past, current and future achievements must be shared with my parents, brother, sister, and my partner, Shannon; for I could accomplish very little without their love and support.

Robert May
Adelaide, 14 October 2009

Contents

Publications

Book chapters

- May, R. J., H. R. Maier, and G. C. Dandy, Development of artificial neural networks for water quality modelling and analysis, in *Modelling of Pollutants in Complex Environmental Systems*, edited by G. Hanrahan, vol. 1, pp. 27–62, ILM Publications, London, UK, 2009.

Journal articles

- May, R. J., G. C. Dandy, H. R. Maier, and T. M. K. G. Fernando, Nonlinear variable selection for artificial neural networks using partial information, *Environmental Modelling and Software*, *23*, 1312–1326, 2008.
- May, R. J., H. R. Maier, G. C. Dandy, and J. B. Nixon, Application of partial mutual information-based variable selection to ANN forecasting of water quality within water distribution systems, *Environmental Modelling and Software*, *23*, 1289–1299, 2008.
- May, R. J., H. R. Maier, and G. C. Dandy, Data Splitting for Artificial Neural Networks Using SOM-based Stratified Sampling, *Neural Networks*, *20*, 283–294, 2010.

Journal articles under review

- May, R. J., H. R. Maier, and G. C. Dandy, SOMPLEX: a hybrid SOM-DUPLEX data splitting algorithm for ANN development, *Submitted to Water Resources Research*

Peer-reviewed conference articles

- May, R. J., G. C. Dandy, H. R. Maier, and T. M. K. G. Fernando, Critical values of a kernel-density based mutual information estimator, in *IEEE International Joint Conference on Neural Networks*, pp. 9997–10,002, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, 2006.

List of Figures

List of Tables

List of Abbreviations

ACF	Auto-correlation function
ACO	Ant colony optimisation
AIC	Akaike Information Criterion
ANN	Artificial neural network
ARMA	Auto-regressive moving-average
ARX	Auto-regressive with exogenous inputs
BIC	Bayesian Information Criterion
BPA	Back-propagation algorithm
CV	Cross-validation
CVI	Cluster validity index
DBS	Density biased sampling
DSS	Data subset selection
EA	Evolutionary algorithm
EANN	Evolutionary neural network
GA	Genetic algorithm
GRIDA	GRNN input determination algorithm
GRNN	Generalised regression neural network
ICA	Independent component analysis
IMC	Inverse model control
IVS	Input variable selection
JMI	Joint mutual information
KDE	Kernel density estimation
MAD	Median absolute deviation from the median
MAE	Mean absolute error
MPC	Model predictive control
MCS	Monte Carlo simulation
MI	Mutual information
MIFS	Mutual information feature selection
MLP	Multi-layer perceptron
MR	Maximum relevance

continued on next page

mR	Minimum redundancy
mRMR	Minimum redundancy–maximum relevance
MRE	Mean relative error
MSE	Mean squared error
PACF	Partial auto-correlation function
PC	Principal component
PCA	Principal component analysis
pdf	Probability density function
PMI	Partial mutual information
PMIS	Partial mutual information-based selection
QE	Quantisation error
RI	Relative importance
RMSE	Root mean squared error
SBSS	SOM-based stratified sampling
SCE	Shuffled complex evolution
SOM	Self-organizing map
SRS	Simple random sampling
SVR	Single variable regression
UVA	Ultra-violet absorbence
VQ	Vector quantisation
WDS	Water distribution system

List of Symbols

General

$a(z_j)$	Activation function
d	Number of variable dimensions
E	Expectation
$f(z_j)$	Hidden node transfer function
$\hat{f}(x), \hat{f}(x,y)$	Density function estimate
$\mathrm{F}(x)$	Process/model transfer function
p	Number of model parameters
r	Coefficient of determination
R	Linear correlation
R'	Partial correlation
V	Variance
W	ANN weight matrix
w_i	ANN weight
X	Random independent/ANN input variable
x	Realisation/observation of X
$\hat{x}$	Model estimate of x
Y	Dependent/ANN output variable
y	Realisation/observation of output variable Y
$\hat{y}$	Model estimate of y
z_j	training input vector (GRNN) / hidden node input (MLP)

Input Variable Selection

B	Bootstrap size
C	Candidate input variable
G	Gaussian kernel function
h	Kernel bandwidth

continued on next page

h_G	Scott reference bandwidth
I	Mutual information
I'	Partial mutual information
k	Number of selected input variables
K_h	Kernel function
$p(x), p(x, y)$	Probability density function
S	Hat matrix defined by $\hat{y} = S(y)$

Data Splitting

$C^{(i)}$	i^{th} cluster
$\mathcal{D}$	Modelling data set
H	Number of strata
k	Number of SOM units $= m \times n$
m	Number of SOM rows
N	Length of dataset D
n	Number of sampled data / number of SOM columns
$n_{\mathcal{S}}$	Number of training data
$n_{\mathcal{T}}$	Number of test data
$n_{\mathcal{V}}$	Number of validation data
S	Silhouette coefficient
$\mathcal{S}$	Test data set
$\mathcal{T}$	Training data set
$\mathcal{V}$	Validating data set

Greek Symbols

ε	Error term
ϕ	Partial auto-correlation
ψ	Golden Ratio ($\sim$1.6)
σ	Standard deviation / GRNN smoothing parameter
μ	Mean

Chapter 1

Introduction

> *"Trying to predict the future is a mug's game. But increasingly it's a game we all have to play because the world is changing so fast and we need to have some sort of idea of what the future's actually going to be like because we are going to have to live there, probably next week."*
>
> Douglas Adams, *The Salmon of Doubt* (2002)

1.1 Research Background

The development of mathematical models of physical and chemical processes is essential for representing real-world environmental systems. Models permit us to experiment and explore system behaviour, which subsequently enables us to increase understanding or make predictions to guide decisions that encompass all aspects such as policy, management, operation, planning and design. In the water resources field, modelling and analysis underpins a range of water resources applications, including prediction and forecasting of water quality in surface waters, rainfall-runoff modelling in catchment hydrology, water and wastewater treatment process modelling and control, and modelling of hydraulics and water quality within water distribution networks.

Model development can follow two distinct approaches: *conceptual*, or *empirical*. The conceptual modelling approach is based on the hypothesises of a mathematical model, which is derived from an understanding or assumption of the nature of the physical processes at work within a system. Conceptual models can be constructed from first principles in a *bottom-up* approach, or by breaking a system into successively smaller components, in a *top-down* approach. For example, water distribution system (WDS) simulation models are widely used based on mathematical terms for pipe hydraulics, and kinetic reaction terms for decay or formation of chemicals within the water. Conceptual models are widely accepted, as the parameters of the model generally have a physical interpretation, and the behaviour of the model can be reconciled with the real-world system. However, the required understanding is not always available, especially in the case of large scale, complex systems. In this case, the number of different mechanisms and their interactions may not be fully understood or known; or, although an adequate conceptual model is available, there may be insufficient data to calibrate the potentially large number of model parameters. Consequently, during conceptual model development, many assumptions or simplification are made, which can often oversimplify complex phenomena, and decrease the accuracy of the model.

The alternative approach to modelling is the empirical or, *inductive* approach, where the model is determined by finding the function that provides the best-fit to a set of data, which comprises observations of system inputs and corresponding outputs. In this case, the goal is simply to find a suitable function to describe the observed behaviour rather than understand it. Classic examples of the empirical approach are multiple linear regression and polynomial regression. The advantage of empirical modelling is that errors introduced by uncertainty, simplifications and assumptions are avoided, and can lead to a more accurate representation of the system, which is more desirable when understanding the processes is less important. Furthermore, the ability to determine an expression from a set of data, without the supposition of a physical mechanism, provides a more expedient approach to model development. A drawback is that the form of the expression is often arbitrary, and therefore the empirical modelling approach is often criticised for having no relationship to the physical process. The model is conventionally referred to as a “black-box”. However, this is not always the case, and it is possible to gain new insight into the behaviour of a previously unknown process by examining the structure of an empirical model that is induced from the data (*Jain et al.*, 2004; *Kingston*, 2006).

Conventional regression and classification techniques have been used in many modelling applications with great success. Linear regression analysis tools, such as the auto-regressive moving-average (ARMA) model, auto-regressive with exogenous inputs (ARX) model, and other similar types of statistical models have been the foundation of time-series forecasting and process system identification for over half a century (*Box and Jenkins*, 1976). However, their main limitation is that many environmental processes are inherently complex and non-linear, and so many of these techniques are found to be inadequate. The recent advent of modern computing, which has provided affordable access to computing power, has allowed more sophisticated statistical modelling paradigms to be explored. Consequently, there has been interest in the application of tools such as artificial neural networks (ANNs) for the development of statistical models of complex environmental processes.

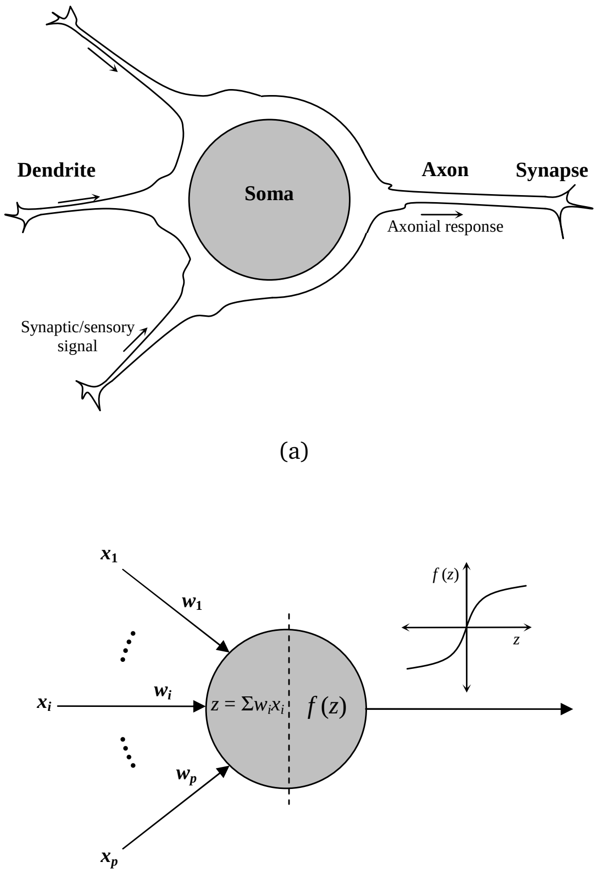
1.2 Artificial Neural Networks

Artificial neural networks (ANNs) are a mathematical modelling paradigm that is inspired by the mechanics of mammalian cognition. The basis for the ANN paradigm is the perceptron, which represents the biological neuron. The biological neuron and its mathematical counterpart are shown in Figure 1.1. In the biological neuron (Figure 1.1(a)), chemical signals are received by the dendrites either from the synapse other neurons, or from sensory cells in response to some external stimulus. The relative strength of each synaptic signal is regulated by the receptors at the terminal end of the dendrites. The combined effect of signals from all dendrites translates to an activation level within the body of the cell (or *soma*). The degree of activation in the soma results in an output signal that is transmitted via the axon to the synapse, which itself may connect to the dendrites of other neurons. The perceptron (Figure 1.1(b)) mathematically represents the biological neuron. Inputs signals are summated to determine an overall stimulus. A transfer function then determines the corresponding degree of activation that results in the output from the perceptron. In this case, the sensitivity to each input signal is determined by corresponding connection weights.

In the case of the perceptron, an input vector X denotes the multiple input signals that are connected to the perceptron by connection weights. The connection weights determine the sensitivity to each input variable. The first component of the perceptron receives the signals, and generates an overall input signal, z, that is the weighted sum of all inputs into the neuron, which is given by:

$$z = \sum_{i=1}^{p} w_i x_i. \tag{1.1}$$

The second component of the perceptron then applies a transfer function, f, to transform the input signal into an activation, or output. Individually, a single perceptron can discriminate one input state from another. However, the immense potential of artificial neural networks (ANNs) is derived from utilising many perceptrons within an interconnected network in a similar fashion to the mammalian



(a)

(b)

Figure 1.1: Biological neuron and the mathematical perceptron. In the biological neuron (a), chemical signals are received by the dendrites either from the synapses of other neurons, or from sensory cells in response to some external stimulus. The relative strength of each synaptic signal is regulated by the receptors at the terminal end of the dendrites. The combined effect of signals from all dendrites translates to an activation level within the body of the cell (or *soma*). The degree of activation in the soma results in an output signal that is transmitted via the axon to the synapse, which itself may connect to the dendrites of other neurons. The perceptron (b) mathematically represents the biological neuron. Inputs signals are summated to determine an overall stimulus. A transfer function then determines the corresponding degree of activation that results in the output from the perceptron. In this case, the sensitivity to each input signal is determined by corresponding connection weights.

brain.

Given the infinite scope for constructing ANNs using this simple basis, many types of ANNs exist. However, generally the following are true about all ANNs:

- *Connectivity*. The structure of an ANN is a connected network of nodes, or *neurons*, that will individually respond to an input signal. The connectionist framework of ANNs defines the mathematical and computational analogy for features of the human brain, and is responsible for the seemingly infinite flexibility and robustness that are often considered to be the most desirable qualities of ANNs.
- *Mapping*. An ANN elicits a global output, based on the interaction between the connected nodes, in response to one or more input stimuli. The ability of ANNs to provide a mapping of input-output relationships is the fundamental basis for their role in tasks such as pattern recognition, signal processing, classification and regression.
- *Adaptation and Learning*. An ANN possesses the ability to adjust its local sensitivity, and hence its global response, based on information that is provided to the network. It is this quality that best defines ANNs as tools for machine learning and artificial intelligence (AI) (*Narendra*, 1991).

In modelling applications, the utility of an ANN is that it can be developed to represent the transfer function F for some otherwise unknown process

$$y = F(x_1, x_2, \ldots, x_d; \mathbf{W}) \tag{1.2}$$

where y is the output variable, and $x_1, \ldots x_d$ represent the set of input variables, and $\mathbf{W}$ denotes the matrix whose elements are the connection weights w_{ij}. Given a set of training data (X, y), the machine learning paradigm is that the ANN can be trained so that the weights W are adjusted to achieve the overall function, F, that most accurately describes the data generating process.

In the case that y takes a categorical value, or *class descriptor*, then the ANN will perform *classification*; and for scalar y, the ANN performs regression. Indeed, many parallels can be drawn between ANN learning and conventional statistical regression, since ANN training is essentially a form of parameter fitting. However, in comparison to conventional regression and classification techniques, ANN models offer several distinct advantages, which are namely:

- *Universal function approximation.* Arguably the most important advantage of using ANN models is that they are capable of mapping any conceivable function, and are therefore capable of performing non-linear regression. This immediately makes them an attractive options for modelling complex processes, which conventional linear regression-based models do not fully describe.

- *Flexible architecture.* Flexibility can be easily incorporated by changing the network architecture, which provides a simple framework for developing an infinite number of different types of network. Additionally, equivalent ANN architectures can be derived for conventional models, including linear regression.

- *Robustness.* ANNs are able to learn relationships with noisy data and the ability to incorporate redundancy allows ANNs to perform classification using either partial or incomplete data. These attributes make ANN models highly suitable for deployment in many real-world applications.

Consequently, there has been growing interest in the application of artificial neural networks (ANNs) to water quality modelling. Many water quality processes are inherently complex and conventional models are often too simplistic to adequately describe their behaviour. In other cases, it is difficult to attempt to determine a functional relationship, since there is an inadequate understanding of the processes involved. For these reasons, modelling practitioners are finding that ANN models provide a suitable solution to these problems. Table 1.1 presents several examples, and illustrates that ANN models are being explored as an approach in a diverse range of applications across the entire field of water resource management.

1.3 Research Objectives

Despite the many potential applications that have been reported, the issue of how to develop ANN models remains an ongoing debate within the research community (*Maier and Dandy*, 2000). A major obstacle in the uptake of ANN modelling is that model development is not transparent, or rigorous. Many of the choices made by ANN modellers tend to be subjective, or not sufficiently justified, which leads to a lack of confidence in the approach. Confounding this issue is that many practitioners in the field of environmental engineering are not experts in the areas of computer science, machine learning, or statistical learning, which leads to the dangerous situation of a little knowledge. This situation often leads to a blind application of ANN tools without an appreciation for the more

Table 1.1: Examples of ANN applications within the water resources field

Reference	Application
Cote et al. (1995)	Modelling an activated sludge process
Rodriguez and Serodes (1996)	Disinfectant residual forecasting
Maier and Dandy (1997)	Water quality time-series forecasting
Brasquet and Le Cloirec (1999)	Quantitative structure-activity relationship (QSAR) for adsorption of chemicals by activated carbon
Damas et al. (2000)	Water supply network modelling and control
Charef et al. (2000)	Remote sensing of chemical oxygen demand
Serodes et al. (2001)	Forecasting disinfectant residuals within a water supply network
Baxter et al. (2001)	Real-time ANN control of coagulation process
Milot et al. (2002)	Predicting disinfection by-product formation in water supply networks
Bowden (2003)	Forecasting cyanobacterial and salinity time-series in rivers
Jain et al. (2004)	Prediction of catchment rainfall-runoff
Cigizoglu (2004)	Prediction of suspended sediments in river water
Zhang (2004)	Real-time forecasting of treated water colour
Maier et al. (2004)	Modelling coagulation jar-tests
Cigizoglu (2005)	Forecasting river flow
Fogelman et al. (2005)	Prediction of chemical oxygen demand with UV spectroscopy
Broad et al. (2005)	Metamodelling for water distribution systems for optimisation speed-up
Kingston (2006)	Forecasting cyanobacteria and salinity in rivers
Alp and Cigizoglu (2007)	Forecasting WTP suspended sediment load
Raduly et al. (2007)	Wastewater treatment plant modelling
May et al. (2008b)	Forecasting disinfectant residual within a water distribution network
Welk (2008)	Forecasting chlorophyll-*a* within an open reservoir

subtle issues relating to ANN model development, which can have a significant (negative) influence on the modelling outcomes (*Sarle*, 1997; *Bowden*, 2003).

This thesis is therefore concerned with the methodology that is applied to the development of ANN models for water quality modelling. In applications that utilise ANN models, 90% of the required effort is during the model development stages (*Morari and Lee*, 1999). It is therefore sensible that efforts are directed to ensuring that (i) the effort required to build ANN models is reduced; (ii) appropriate techniques are employed, so that the ANN developed makes the best use of the data that are available; and (iii) that the development methodology be made as transparent as possible to ensure confidence in the models.

A proposed framework for ANN model development is illustrated in Figure 1.2 (*May et al.*, 2009a), which describes each of the stages from data collection through to model deployment. The most important aspect to note is that there are several important stages of model development either side of ANN training. It is important to equally consider each stage of ANN development, since the quality of the stages preceding ANN training will affect the quality of training, and the performance of the resulting model. This framework is consistent with statistical learning theory, which is as equally concerned with aspects of necessary data transformations (pre-processing), variable selection and data splitting, as it is with parameter estimation. A more detailed summary of the considerations at each step is given in Chapter 10, however the stages can be briefly summarised as follows:

1. *Data collection.* A common step in the development of *any* model is the observation and measurement of variables to generate a set of input and output data that describe the behaviour of the system. Data collection may use existing historical databases, or may require monitoring especially for the purpose of model development. Where possible, experimental manipulation of the system can also provide datasets that represent a wider range of system behaviour.

2. *Data pre-processing.* This stage refers to the necessary cleansing of raw data to remove errors and in-fill missing data. It may also include mathematical transformations such as smoothing to a consistent time-interval, scaling to a consistent range and generating time-series delays, as required.

3. *Input variable selection.* This step involves the screening of input variables to identify and eliminate redundant or irrelevant variables to achieve the most parsimonious set of input variables (i.e. maximum information with the minimum number of variables).

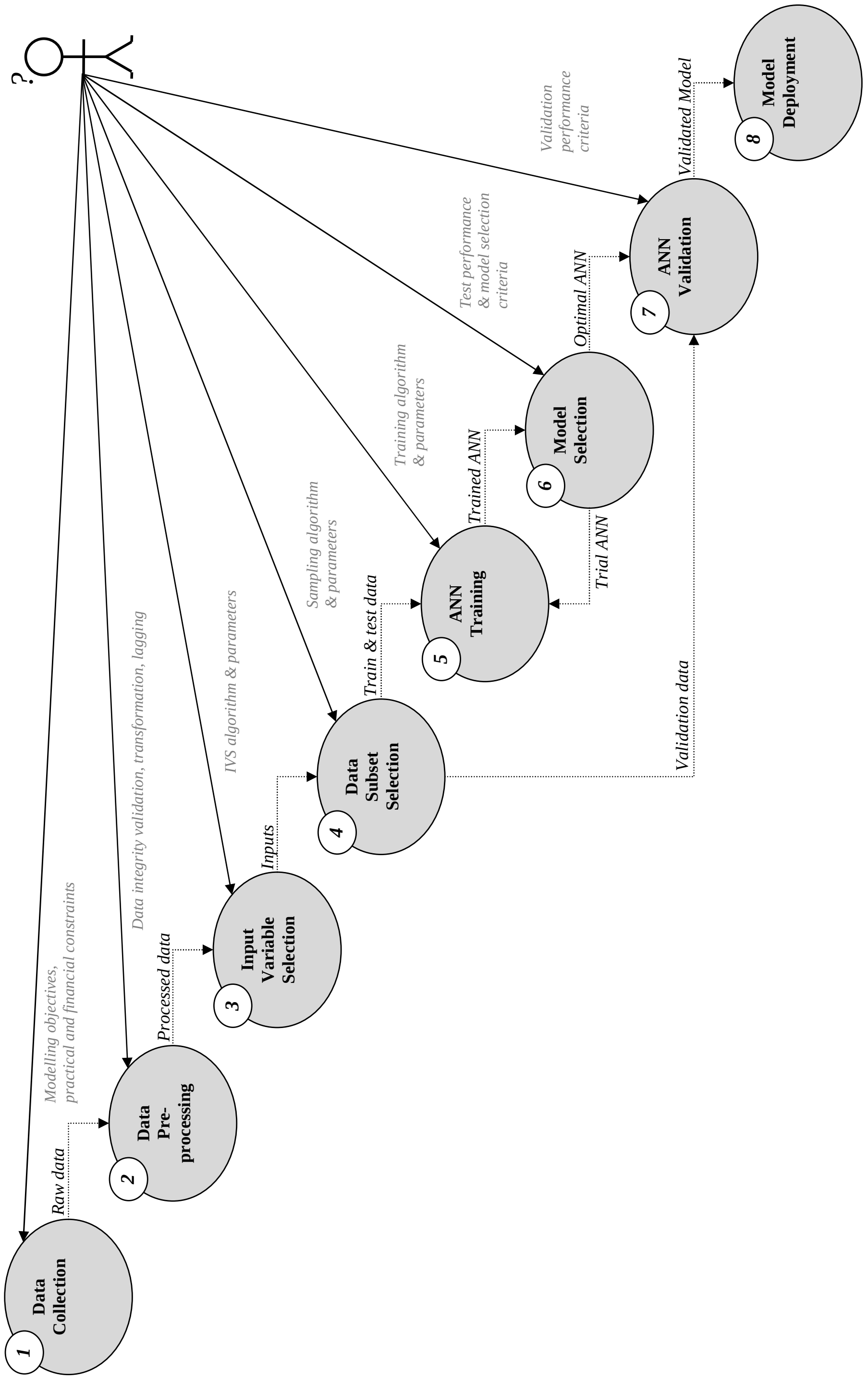

?
1
Data Collection
Raw data
Modelling objectives, practical and financial constraints
2
Data Pre-processing
Processed data
Data integrity validation, transformation, lagging
3
Input Variable Selection
Inputs
IVS algorithm & parameters
4
Data Subset Selection
Train & test data
Sampling algorithm & parameters
Validation data
5
ANN Training
Trained ANN
Training algorithm & parameters
Trial ANN
6
Model Selection
Optimal ANN
Test performance & model selection criteria
7
ANN Validation
Validated Model
Validation performance criteria
8
Model Deployment

Figure 1.2: Proposed framework for ANN model development.

4. *Data splitting.* In any model development, at least two data sets are required: calibration and validation. Because only one dataset is available for model development, splitting the modelling data is necessary to create these independent datasets. However, in ANN model development, the calibration data proportion is generally further divided into training and test portions, so that three independent samples are generated.

5. *Training.* The ANN learns the optimal mapping of inputs to outputs through training. During training, the initially random connection weights are iteratively changed, according to the chosen training algorithm, in response to the prediction error for training observations. This stage is analogous to parameter estimation in regression, or calibration of conceptual model parameters. Since an ANN can be trained to eventually fit the training data perfectly, training is generally terminated based on optimal performance on the test data, so that the ANN adequately generalises. Hence, the requirement for the additional set of test data.

6. *Model Selection.* Since the optimal form of an ANN (e.g. number of hidden nodes) is unknown, it is often necessary to compare ANNs with different architectures to fine tune the degree of complexity. The application of model selection criteria identifies the best performing ANN architecture based on test performance, and considers trade-offs between model complexity and accuracy, which forms an important aspect of model assessment in statistical modelling.

7. *Validation.* This stage includes the validation of ANN predictive performance on previously independent data not used in the training or test stages, to provide an assessment of the model's likely performance when deployed. This is also a required step during the development of any type of model: statistical, or physical.

8. *Model deployment.* Once the model is validated, it is ready for deployment in its intended application. Since the model is initially constructed using a limited sample of data, it is often useful to retrain the model as novel data become available; which are not represented in the original set of data collected for model development. This is similar to re-calibration, and can ensure that model accuracy does not degrade over a long-term deployment.

Methodologies for undertaking the training stage (Stage 5) of development—ANN architectures and learning algorithms—have been investigated extensively within the literature, although interestingly this forms just one of eight steps in model development. This is not surprising, given that the greater proportion of literature on ANNs is written from a machine learning and cognition perspective, rather than from a statistical learning perspective. However, the latter is more appropriate when considering that the intended application is to develop a statistical model for regression or classification. The same issue is also reflected in the functionality of popular software tools that implement ANN development, but which do not always adopt state-of-the-art methods. Many practitioners and researchers lack the necessary skills or resources to develop ANN modelling tools, and rely instead on readily available tools to learn, teach or apply ANN modelling techniques. Consequently, many of the ANN modelling paradigms commonly employed are influenced by the limitations and restrictions of current software. In comparison to ANN architectures and learning algorithms, the stages of input variable selection (IVS) (Stage 3) and data splitting (Stage 4) tend to be considered relatively less in the majority of reported ANN applications. Although it considered by most practitioners that these are necessary steps during ANN development, it is how to perform these tasks, which remains most unclear.

Recent interest in methods for IVS that are applicable for ANN development has resulted in a variety of different algorithms proposed within the literature. However, few comparisons of different ANN approaches are made that consider the benefits or similarities of each approach, making it difficult to determine the most appropriate approach. As will be discussed, the development of IVS methods based on estimation of mutual information, which is based on information theory, has become a popular focus of ANN research. Using this relatively new approach seems quite appropriate for ANN development, but potential improvements on current algorithm designs are necessary to address reliability and computational performance limitations. Consequently, the specific objectives of this research, with respect to the issue of input variable selection, are to:

1. Comprehensively review IVS algorithms for ANN development,
2. Provide a benchmark comparison of MI-based approaches against conventional correlation,
3. Further develop the use of mutual information for input variable selection, and
4. Validate the IVS approach and demonstrate its benefit for real-world examples.

Data splitting, which is the fourth stage of the framework in Figure 1.2 is an important aspect of ANN development. Not only are independent training and test data required for ensuring trained ANN models can adequately generalise, but a separate set of data must also be available to validate ANN performance. Data splitting is the most widely adopted approach for generating three representative samples from the available modelling dataset. However, as is discussed in Chapter 3, data splitting is often poorly implemented, which can have a significant impact on the subsequent stages of training, model selection and model validation. The case against randomly splitting data—the approach that is most commonly employed—is evident, although it needs to be reinforced. A number of alternative algorithms have been proposed within the literature to overcome the issues with random sampling. In particular, an approach using the self-organizing map (SOM) has been suggested. However, there is no consensus on how this approach should specifically be implemented to achieve the best results. Consequently, the specific objectives of this research, with respect to data splitting, are to:

1. Perform a broad review and comparison of sampling algorithms suitable for data splitting,
2. Identify a suitable definition and methodology for quantifying data splitting quality,
3. Further develop the methodology for data splitting using the SOM,
4. Compare the performance of different approaches from the perspective of sample quality,
5. Determine guidelines for selecting the most appropriate data splitting algorithm, and
6. Validate the outcomes through the application to real-world examples.

1.4 Thesis Structure

This thesis is presented as a collection of articles, which have arisen from the research undertaken. The contents of the thesis are logically ordered to first present all of the necessary background, and the then present each of the articles. Chapters 2 and 3 provide comprehensive reviews of literature pertaining to input variable selection and data subset selection, respectively. These reviews provide a basis for the arguments that are later presented in the published work. It is worthwhile providing such reviews, as although many articles have been published in this area, there are relatively few that have critically reviewed and compared the wide range of different approaches that have been reported for these two important components of the ANN development framework.

Chapter 4 provides a synopsis of the publications that subsequently form the remainder of this thesis, and which present the core of the research undertaken. The synopsis provides a summary of the contributions of each individual article to provide an overall cohesive context. Six publications are presented as chapters Chapter 5 through 10. The first three publications are concerned with the issue of input variable selection. The first two present the theory relating to the development of a novel algorithm, and the third presents the application of the algorithm within an applied context. Chapter 8 and 9 present the fourth and fifth papers, which are concerned with the subject of data splitting. The fourth paper discusses the issues surrounding the application of the self-organizing map (SOM) to data splitting. The fifth paper presents a novel data splitting algorithm called SOMPLEX, and compares its performance with a range of approaches for a set of water resources case studies. Finally, Chapter 10 presents the sixth article, which is published book chapter that summarises the state-of-the-art of ANN development and application.

Concluding remarks summarising the key contributions of the thesis, discussion on limitations and recommendations for future research directions are given in Chapter 11.

Chapter 2

Input Variable Selection

> *"If variable elimination has not been sorted out after two decades of work assisted by high-speed computing, then perhaps the time has come to move on to other problems."*
>
> R. L. Plackett, discussion in *Miller* (1984)

2.1 Introduction

The choice of input variables is a fundamental, and yet crucial consideration in identifying the optimal functional form of statistical models. The task of selecting input variables is common to the development of all statistical models, and is largely dependent on the discovery of relationships within the available data to identify suitable predictors of the model output. In the case of parametric, or semi-parametric empirical models, the difficulty of the input variable selection task is somewhat alleviated by the *a priori* assumption of the functional form of the model, which is based on some physical interpretation of the underlying system or process being modelled. However, in the case of ANN and other similarly data-driven statistical modelling approaches, there is no such assumption made regarding the structure of the model. Instead, the input variables are selected from the available data, and the model is developed subsequently. The difficulty of selecting input variables arises due to (i) the number of available variables, which may be very large; (ii) correlations between potential input variables, which creates redundancy; and (iii) variables that have little or no predictive power.

Variable subset selection has been a longstanding issue in fields of applied statistics dealing with inference and linear regression (*Miller*, 1984), and the advent of ANN models has only served to create new challenges in this field. The non-linearity, inherent complexity and non-parametric nature of ANN regression make it difficult to apply many existing analytical variable selection methods. The difficulty of selecting input variables is further exacerbated during ANN development, since the task of selecting inputs is often delegated to the ANN during the learning phase of development. A popular notion is that an ANN is adequately capable of identifying redundant and noise variables during training, and that the trained network will use only the salient input variables. ANN architectures can be built with arbitrary flexibility and can be successfully trained using any combination of input variables (assuming they are good predictors). Consequently, allowances are often made for a large number of input variables, with the belief that the ability to incorporate such flexibility and redundancy creates a more robust model. Such pragmatism is perhaps symptomatic of the popularisation of ANN models through machine learning, rather than statistical learning theory. ANN models are too often developed without due consideration given to the effect that the choice of input variables has on model complexity, learning difficulty, and performance of the subsequently trained ANN.

The following review presents the IVS problem within the context of ANN model development for time-series forecasting and function approximation applications, such as are typically encountered by environmental modellers. Although the need to adopt a methodical approach to IVS for ANN development is well-justified, the importance of the task is not as well recognised by environmental modellers as it is elsewhere (*Maier and Dandy*, 2000; *Bowden*, 2003). This is evident from the myriad of methods that are employed to undertake the IVS task within other reported ANN modelling applications. Consequently, the first part of this chapter presents a comprehensive review of IVS approaches that have been developed to address the issue, which draws from recent innovations that have been presented in fields such as signal processing, pattern recognition, gene expression data analysis and classification.

2.2 The Input Variable Selection Problem

Recall that for an unknown, steady-state input-output process, the development of an ANN provides the non-linear transfer function

$$Y = F(X) + \varepsilon, \tag{2.1}$$

where the model output Y is some variable of interest, X is a k-dimensional input vector, whose component variables are denoted by $X_i (i = 1, \ldots, k)$, and ε

is some small random noise. Let C denote the set of d variables that are available to construct the ANN model. The I_{d-k} problem of input variable selection (IVS) is to choose a set of k variables from C to form X (*Kwak and Choi*, 2002; *Battiti*, 1994) that leads to the optimal form of the model, F.

Dynamic processes will require the development of an ANN to provide a time-series model of the general form

$$Y(t+k) = F(Y(t), \ldots, Y(t-p), X(t), \ldots, X(t-p)) + \varepsilon(t). \tag{2.2}$$

Here, the output variable is predicted at some future time $t + k$, as a function of past values of both input X and output Y. Past observations of each variable are referred to as *lags*, and the model order p defines the maximum lag of the model. The model order reflects the persistence of dynamics within the system. In comparison to the steady-state model formulation, the number of variables in the candidate set C is now multiplied by the model order. Consequently, for systems with strong persistence, the number of candidate variables is often quite large.

ANN models may be specified with insufficient, or uninformative input variables (under-specified); or more inputs than is strictly necessary (over-specified), due to the inclusion of superfluous variables that are uninformative, weakly informative, or redundant. Defining what constitutes an optimal set of ANN input variables first requires some consideration of the impact that the choice of input variables has on model performance. The following arguments summarise the key considerations:

- *Relevance.* Arguably the most obvious concern is that too few variables are selected, or that the selected set of input variables is not sufficiently informative. In this case, the outcome is a poorly performing model, since some of the behaviour of the output remains unexplained by the selected input variables. In most cases, it is reasonable to assume that a modeller will have some expert knowledge of the system under consideration; will have surveyed the available data, and will have arrived at a reasonable set of candidate input variables. The *a priori* assumption of model development is that at least one or more of the available candidate variables is capable of describing some, if not all, of the output behaviour, and that it is the nature and relative strength of these relationships that is unknown (which is, of course, the motivation behind the development of non-parametric models). Should it happen that none of the available candidates are good predictors, then the problem of model development is intractable, and it may be necessary to reconsider the available data and the choice of model output, and to undertake further measurements or observations before revisiting the task of model development.

- *Computational Effort.* The immediately obvious effect of including a greater number of input variables is that the size of an ANN increases, which increases the computational burden associated with querying the network—a significant influence in determining the speed of training. In the case of the MLP, the input layer is likely to have an increased number of nodes, and in a fully connected network, the number of connection weights can increase dramatically, since each additional input adds another connection to each of the j nodes in the first hidden layer (excluding the bias node). In the case of GRNN and RBF networks, the computation of distance to prototype vectors is more expensive due to higher dimensionality. Furthermore, additional variables place an increased burden on any data pre-processing steps that may be undertaken during ANN development.

- *Training difficulty.* The task of training an ANN becomes more difficult due to the inclusion of redundant and irrelevant input variables. The effect of redundant variables is to increase the number of local optima in the error function that is projected over the parameter space of the model, since there are more combinations of parameters that can yield locally optimal error values. Algorithms such as the back-propagation algorithm, which are based on gradient descent, are therefore more likely to converge to a local optimum resulting in poor generalisation performance. Training of the network is also slower because the relationship between redundant parameters and the error is more difficult to map. Irrelevant variables add noise into the model, which also hinders the learning process. The training algorithm may expend resources adjusting weights that have no bearing on the output variable, or the noise may mask the important input-output relationships. Consequently, many more iterations of the training algorithm may be required to determine a near-global optimum error, which adds to the computational burden of model development.

- *Dimensionality.* The so-called *curse of dimensionality* (*Bellman*, 1961) is that, as the dimensionality of a model increases linearly, the total volume of the modelling problem domain increases exponentially. Hence, in order to map a given function over the model parameter space with sufficient confidence, an exponentially increasing number of samples is required (*Scott*, 1992). Alternatively, where a finite number of data are available (as is generally the case in real-world applications), it can be said that the confidence or certainty that the true mapping has been found will diminish. ANN architectures like the MLP

Table 2.1: Growth of sample size with increasing dimensionality required to maintain a constant standard error of the probability of an input estimated in the GRNN pattern layer (*Silverman*, 1986).

Dimension, d	Sample size, N
1	4
2	19
3	67
4	223
5	768
6	2790
7	10 700
8	43 700
9	18 700
10	842 000

are particularly susceptible to the curse due to the rapid growth in the number of connection weights as input variables are added. Table 2.1 illustrates the growth in the sample size required to maintain a constant error associated with estimates of the input probability, as determined by the pattern layer of a GRNN. Some ANN architectures can also circumvent the curse of dimensionality through their handling of redundancy and their ability to simply ignore irrelevant variables (*Sarle*, 1997). Others, such as RBF networks and GRNN architectures, are unable to achieve this without significant modifications to the behaviour of their kernel functions, and are particularly sensitive to increasing dimensionality (*Specht*, 1991).

- *Comprehensibility.* In many applications, such as in the case of ANN transfer functions for process modelling, it will often suffice to regard an ANN as a "black-box"' model. However, ANN modellers are increasingly concerned with the development of ANN models for knowledge discovery from data (KDD) and data mining (*Craven and Shavlik*, 1998). The goal of KDD is to train an ANN based on observations of a process, and then interrogate the ANN to gain further understanding of the process behaviour it has learned. Rule-extraction from ANN models can be useful for a number of purposes, including: (i) defining input domains that produce certain ANN outputs, which can be useful knowledge in itself; (ii) validation of the ANN behaviour (e.g. verifying that input-output response trends make sense), which increases confidence in the ANN predictions; and (iii) the discovery of new relationships, which reveals previously unknown insights into the underlying physical process (*Craven and Shavlik*, 1998; *Darbari*, 2000). Reducing the complexity of the ANN architecture, by minimising redundancy and the size of the network, can significantly improve the performance of data mining and rule extraction algorithms.

Based on the arguments presented, a desirable input variable is a highly informative explanatory variable (i.e a good predictor) that is dissimilar to other input variables (i.e. independent). Consequently, the optimal input variable set will contain the fewest input variables required to describe the behaviour of the output variable, with a minimum degree of redundancy and with no uninformative (noise) variables. Identification of an optimal set of input variables will lead to a more accurate, efficient, cost-effective and easily interpretible ANN model.

The fundamental importance of the IVS issue is evident from the depth of literature surrounding the development and discussion of IVS algorithms in fields such as classification, machine learning, statistical learning theory, and many other fields where ANN models are applied. In a broad context, reviews of IVS approaches have been presented by *Kohavi and John* (1997), *Blum and Langley* (1997) and more recently, by *Guyon and Elisseeff* (2003). However, in many examples of the application of ANNs to environmental modelling and data analysis applications, the importance of IVS is often understated. In other cases, the task is given only marginal consideration and this often results in the application of *ad hoc* or inappropriate methods. Reviews by *Maier and Dandy* (2000) and *Bowden* (2003) examined the IVS methods that have been applied to ANN applications in water engineering and concluded that there was a need for a more considered approach to the IVS task. Certainly, no consensus has been reached regarding suitable methods for undertaking the IVS task in the development of ANN regression or time-series forecasting models (*Bowden*, 2003).

2.3 Strategies and Algorithms

A broader review of relevant literature reveals that numerous approaches have been described for undertaking IVS, including a wide range of algorithms for automating the IVS task. The IVS problem has been an ongoing area of research that has evolved based on regression, statistical learning theory, and more recently machine learning. Figure 2.1 presents a taxonomy of the various approaches that has been defined on the basis of the literature review.

It should be noted that the IVS problem is synonymous with feature selection, variable selection, feature extraction, dimensionality reduction etc. Often the differences in these applications are simply the nomenclature and other conventions that are influenced by the field in which they are discussed (i.e. statistics or machine learning). In this thesis, a distinction is made between feature selection and variable selection. The notion of feature selection in classification considers datasets where classes of objects are defined by attributes or features, which take on discrete numeric or categorical values. Many algorithms for feature selection have also been described, however this thesis is focused on algorithms that are applicable to datasets of continuous variables, for which some feature selection approaches are not directly applicable.

IVS algorithms can be broadly classified into two main classes: *wrapper* or *filter* algorithms (*Kohavi and John*, 1997; *Blum and Langley*, 1997), as shown in Figure 2.1. The two main conceptual approaches to IVS algorithm design are illustrated in Figure 2.2. Wrapper algorithms, as shown in Figure 2.2(a), approach the IVS task as part of the optimisation of model architecture. The optimisation searches through the set, or a subset, of all possible combinations of input variables, and selects the set that yields the optimal generalisation performance of the trained ANN. In contrast, IVS filters (Figure 2.2(b)) distinctly separate the IVS task from ANN training and instead adopt an auxiliary statistical analysis technique to measure the relevance of individual, or combinations of, input variables.

Given the general basis for the formulation of both IVS wrapper and filter designs, the diversity of implementations that can possibly be conceived is immediately apparent. However, designs for wrappers and filters share the same overall components, in that, in addition to a measure of the informativeness of input variables, each class of selection algorithms requires:

- an optimality criterion to determine when the optimal set of input variables has been selected, and
- a strategy for searching through the available candidates.

2.3.1 Optimality Criteria

The optimality criterion defines the interpretation of the arguments presented in Section 2.2 into an expression for the optimal size k and composition of the input vector, X. Optimality criteria for wrapper selection algorithms are derived from, or are exactly the same as, criteria that are ultimately used to assess the predictive performance of the trained ANN. Essentially, the wrapper approach treats the IVS task as a model selection exercise, where each model corresponds to a unique combination of input variables. Recall that the most commonly adopted measure

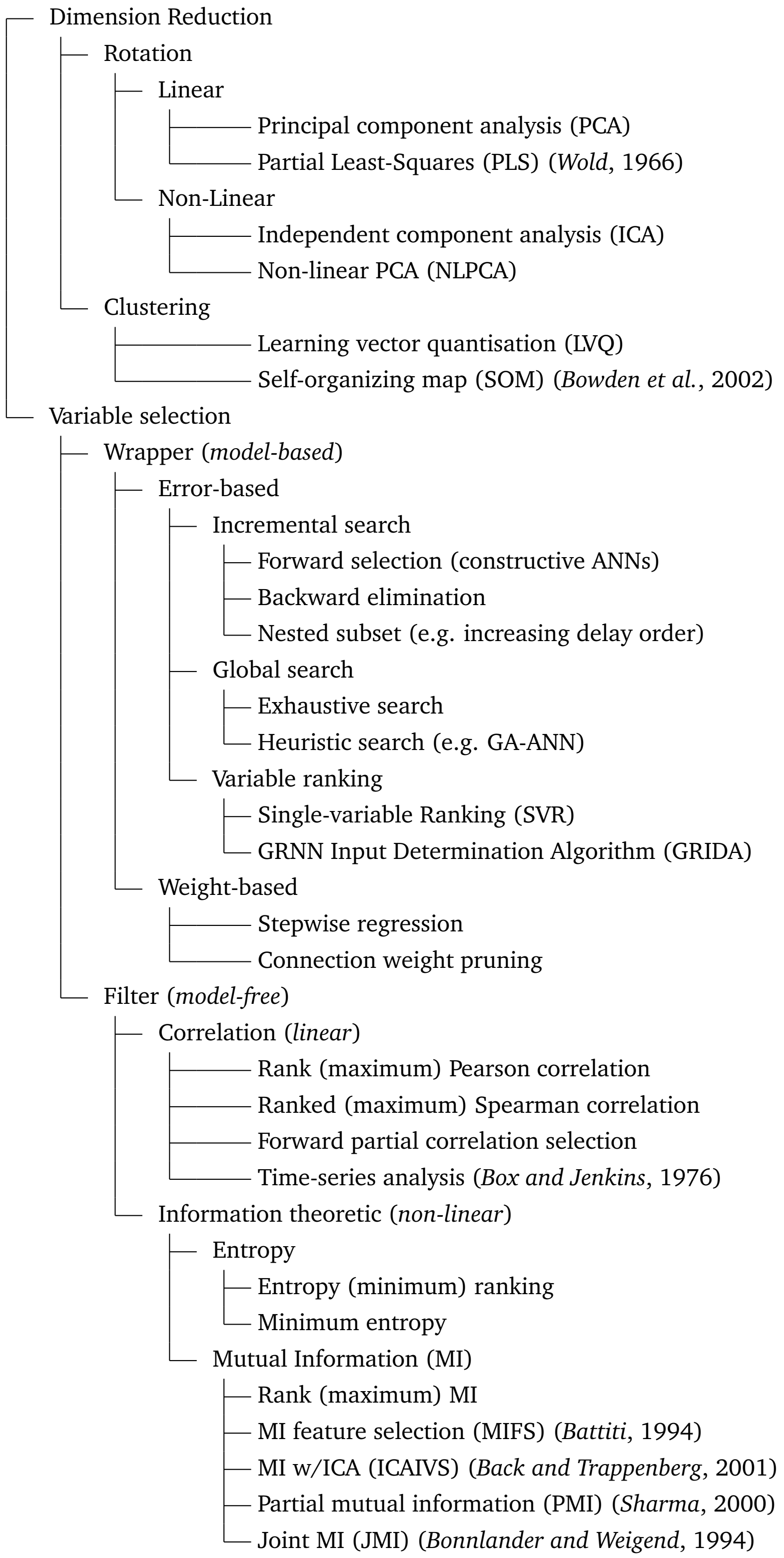
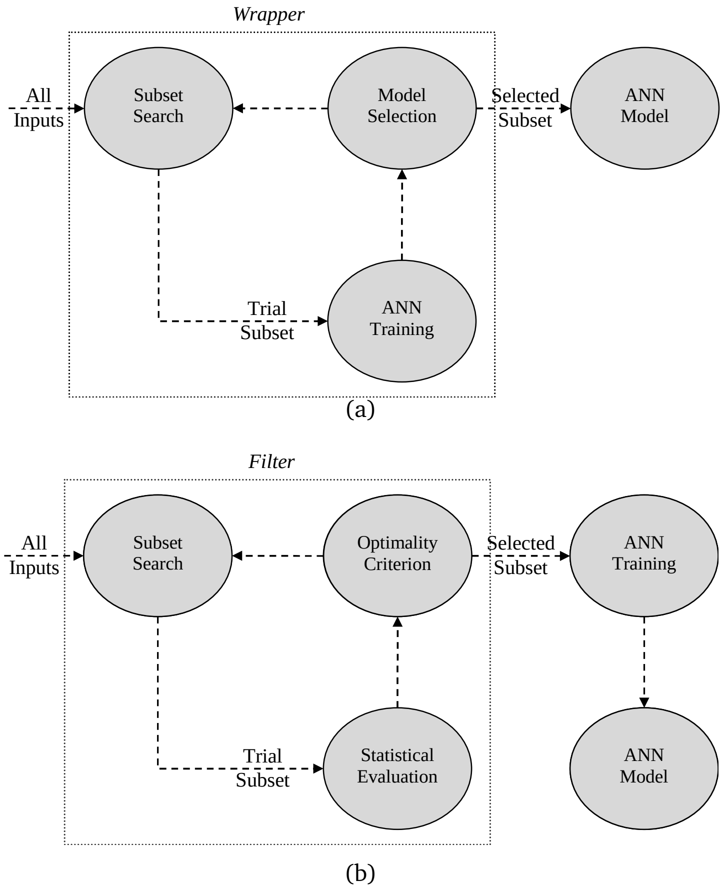

Dimension Reduction
Rotation
Linear
Principal component analysis (PCA)
Partial Least-Squares (PLS) (*Wold*, 1966)
Non-Linear
Independent component analysis (ICA)
Non-linear PCA (NLPCA)
Clustering
Learning vector quantisation (LVQ)
Self-organizing map (SOM) (*Bowden et al.*, 2002)
Variable selection
Wrapper (*model-based*)
Error-based
Incremental search
Forward selection (constructive ANNs)
Backward elimination
Nested subset (e.g. increasing delay order)
Global search
Exhaustive search
Heuristic search (e.g. GA-ANN)
Variable ranking
Single-variable Ranking (SVR)
GRNN Input Determination Algorithm (GRIDA)
Weight-based
Stepwise regression
Connection weight pruning
Filter (*model-free*)
Correlation (*linear*)
Rank (maximum) Pearson correlation
Ranked (maximum) Spearman correlation
Forward partial correlation selection
Time-series analysis (*Box and Jenkins*, 1976)
Information theoretic (*non-linear*)
Entropy
Entropy (minimum) ranking
Minimum entropy
Mutual Information (MI)
Rank (maximum) MI
MI feature selection (MIFS) (*Battiti*, 1994)
MI w/ICA (ICAIVS) (*Back and Trappenberg*, 2001)
Partial mutual information (PMI) (*Sharma*, 2000)
Joint MI (JMI) (*Bonnlander and Weigend*, 1994)

Figure 2.1: Taxonomy of IVS Strategies and Algorithms



(a)

(b)

Figure 2.2: Conceptual IVS algorithm based on (a) a wrapper and (b) filter design.

of predictive performance for ANNs is the mean squared error (MSE), which is given by

$$\mathrm{MSE} = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y_j})^2 \tag{2.3}$$

where y_j and $\hat{y_j}$ are the actual and predicted outputs, which correspond to a set of test data. Following the development of m models, a simple strategy is to select the model that corresponds to the minimum MSE. However, the drawback of this criterion is that the "best"' performing model, in terms of the MSE, is not necessarily the "optimal"' model, since models with a large number of input variables tend to be biased as a result of over-fitting. Consequently, it is more common to adopt an optimality criterion such as Mallows' C_p (*Mallows*, 1973), or the Akaike information criterion (AIC) (*Akaike*, 1974), which penalise overfitting. Both Mallows' C_p and the AIC determine the optimal number of input variables by defining the optimal trade-off between model size and accuracy by penalising models with an increasing number of parameters. In fact, the C_p criterion is considered to be a special case of the AIC.

Mallows' C_p is is defined as

$$C_p = \frac{\sum_{j=1}^{n}(y_j - \hat{y_j}(k))^2}{\sigma_d^2} - n + 2p, \tag{2.4}$$

where $y_j(k)$ are the outputs generated by a model using p parameters, and σ_d^2 are residuals for a full model trained using all d possible input variables. C_p measures the relative bias and variance of a model with p variables. The theoretical value of C_p for an unbiased (optimal) model will be p, and in model selection, the model with the C_p value that is closest to p is selected.

The AIC is defined as

$$\mathrm{AIC} = -n\log\frac{\sum_{j=1}^{n}(y_j - \hat{y_j}(k))^2}{n} + 2(p+1). \tag{2.5}$$

Here, the accuracy is determined by the log-likelihood, which is a function of the MSE. The complexity of the model is determined by the term $p+1$, where p is the number of model parameters. Typically, the regression error decreases with increasing p, but since the model is more likely to be over-fit for a fixed sample size, the increasing complexity is penalised. At some point an optimal AIC is determined, which represents the optimal trade-off between model accuracy and model complexity. The optimum model is determined by minimising the AIC with respect to the number of model parameters, p.

Other model selection criteria have also been similarly derived, such as the Bayesian information criterion (BIC) (*Schwarz*, 1978), which is similar to the AIC, although it applies a more severe penalty of $(k \ln n)$ to the number of model parameters. The expression for the AIC in (2.5) assumes a linear regression model, but can be extended to non-linear regression. However, it should be noted that in this case, $p + 1$ no longer sufficiently describes the complexity of the model and other measures are required. Such measures include the *effective number of parameters*, or Vapnik-Chernovenkis dimension. The values of these measures are a function of the class of regression model that is estimated and the training data. The effective number of parameters, d can be determined by trace(S), where S is a matrix defined by the expression

$$\hat{y} = Sy. \tag{2.6}$$

For kernel regression, the hat matrix, S, is equal to $K^T K$, where the elements of K correspond to each $K_j(x, h)$, and the complexity is therefore given by trace$(K^T K)$. Factors affecting complexity include the number of data, the dimension of the data, and the number of basis functions. The VC-dimension is similarly defined as the number of data points that can be *shattered* by the model (i.e. how many points in space can be uniquely separated by the regression function). However, calculating the VC-dimension of complex regression functions can be difficult (*Hastie et al.*, 2001). For MLP architectures, the VC-dimension is related to the number of connection weights, and for RBF networks the VC-dimension depends on the number of basis functions and their respective bandwidths, if different value is used for each basis function. Both the effective number of parameters and the VC-dimension revert to the value of $p+1$ for linear models.

In filter algorithm designs, the optimality criterion is embedded in the statistical analysis of candidate variables, which defines the interpretation of "good"' input variables. In general, selection filters search amongst the candidate variables and identify suitable input variables according to the following criteria:

- maximum relevance (MR),
- minimum redundancy (mR), and
- minimum redundancy–maximum Relevance (mRMR).

The criterion of maximum relevance ensures that the selected input variables are highly informative by searching for variables that have a high degree of correlation with the output variable. Input ranking schemes are a prime example of MR techniques, in which the relevance is determined for each input variable with the output variable. Greedy selection can be applied to select the k most relevant variables, or a threshold value can be applied to select inputs that are relevant, and reject those which are not.

The issue with MR criteria is that the selection of the k most relevant candidate variables does not strictly yield an optimal ANN. Here, *Kohavi and John* (1997) make the distinction between relevance and usefulness by observing that redundancy between variables can render highly relevant variables useless as predictors. Consequently, a criterion of minimum redundancy aims to find inputs that are maximally dissimilar from one another, in order to select the most useful set of relevant variables. The application of an additional mR criterion with the existing MR criterion leads to mRMR selection criteria, where input variables are evaluated with the dual consideration of relevance, with respect to the output variable; and independence (dissimilarity), with respect to the other candidate variables (*Ding and Peng*, 2005).

2.3.2 Search Strategies

Search strategies applied to IVS algorithms seek to provide an efficient method for searching through the many possible combinations of input variables and determining an optimal, or near optimal set, while working within computational constraints. Searches may be global, and consider many combinations; or local methods, which begin at a start location and move through the search space incrementally. The latter are also commonly referred to as *nested subset* techniques, since the region they explore comprises overlapping (i.e. nested) sets by incrementally adding variables.

Exhaustive Search

Exhaustive search simply evaluates all of the possible combinations of input variables and selects the best set according to the predetermined optimality criteria. The method is the only selection technique that is guaranteed to determine the optimal set of input variables for a given ANN model (*Bonnlander and Weigend*, 1994). Given the combinatorial nature of the IVS problem, the number of possible subsets that form the search space is equal to 2^d, with subsets ranging in size from single input variables to the set of all available input variables. Exhaustive evaluation of all of these possible combinations may be feasible when the dimensionality of the candidate set is low, but quickly becomes infeasible as dimensionality increases.

Heuristic Search

Heuristic search techniques are widely used in optimisation problems where the search space is large. Heuristic search algorithms are particularly adept at efficiently finding global, or near-global optimum solutions within large search spaces by exploiting the common attributes of good solutions. In general, the various algorithms each implement a search that combines random evaluation of solutions throughout the entire search space, with a mechanism to increase the focus of the search in regions that lead to good solutions. Examples of heuristic search algorithms applied to IVS include evolutionary algorithms (EAs), such as genetic algorithms (GAs) (*Bowden*, 2003) and ant colony optimization (ACO) (*Izrailev and Agrafiotis*, 2002; *Marcoulides and Drezner*, 2003; *Shen et al.*, 2005).

GAs are a sub-class of evolutionary algorithms that are inspired by natural evolutionary mechanisms such as breeding (crossing), mutation and selection (*Goldberg*, 1989). In a basic GA wrapper formulation, the decision to include a candidate variable may be encoded as a binary digit 1 (included), or 0 (excluded), so that each possible input set is represented by a string (chromosome) of d digits. Starting with an initial random population represented by m chromosomes, an optimal set of input variables naturally evolves through iterative evaluation of the model error associated with each chromosome, removal of bad solutions, and formulating new chromosomes by crossing previously good ones. The objective function for evaluation of chromosomes is minimisation of the cross-validation error of the trained ANN.

ACO algorithms are a search technique based on the ability of ants to collectively determine the optimal (shortest) pathway to gather resources, such as food

(*Dorigo and Sutzle*, 2004). In the formulation of an IVS search strategy, ACO algorithms define the possible input variable sets as pathways formed by the successive decision to include or exclude each candidate variable (*Marcoulides and Drezner*, 2003). Ant pheromone levels associated with each pathway define the probability of selecting a given path (i.e. combination of input variables), which is initially equal. Pheromone is increased along pathways that yield good input variable sets (as evaluated by cross-validation). The probability of selecting good pathways increases so that, over time, an optimal solution iteratively evolves as the pathway with the highest pheromone level.

The application of heuristic optimisation techniques to IVS wrapper design overcomes the significant computational requirement of exhaustive search, while maintaining the desirable characteristic of providing a global (or, near-global) optimum. Moreover, EA-based IVS wrappers are an attractive option because they can also be included as part of evolutionary ANN training algorithms, which also seek to determine optimal ANN parameter values by minimising the ANN cross-validation error. However, the application of heuristic search techniques requires calibration of search algorithm parameters, which is itself not a trivial task. In general, setting the search parameters involves a trade-off between the amount the search space that is explored, and the rate at which the algorithm converges to a final solution. Finally, heuristic algorithms retain a certain degree of randomness, and although they search more solutions in comparison to sequential selection algorithms, there is still no guarantee that the sub-space explored will include the globally optimal solution.

Forward Selection

Forward selection is a linear incremental search strategy that selects individual candidate variables one at a time. In the case of wrappers, the method starts by training d single-variable ANN models and selecting the input variable that maximises the model performance-based optimality criterion. Selection then continues by iteratively training $d-1$ bivariate ANN models, in each case adding a remaining candidate to the previously selected input variable. Selection is terminated when the addition of another input variable fails to improve the performance of the ANN model. In filter designs, the single most relevant candidate variable is selected first, and then forward selection proceeds by iteratively identifying the next most relevant candidate and evaluating whether the variable should be selected, until the optimality criterion is satisfied.

The approach is computationally efficient overall, and tends to result in the selection of relatively small input variable sets, since it considers the smallest possible models, and trials increasingly larger input variable sets until the optimal set is reached. However, because forward selection does not consider all of the possible combinations, and only searches a small subset, it is possible that the algorithm may encounter a locally optimum set of input variables and terminate prematurely. Also, due to the incremental nature of the forward search, the algorithm may ignore highly informative combinations of input variables that are only marginally relevant individually (*Guyon and Elisseeff*, 2003).

Step-wise Selection

Forward selection is said to have fidelity, in that once an input variable is selected, the selection can not be undone. Step-wise selection is an extension of the forward selection approach, however, input variables may also be removed at any subsequent iteration. The formulation of the step-wise approach is aimed at handling redundancy between candidate variables. For example, a variable X_a may be selected initially due to high relevance, but is later found to be inferior to the combination of two other variables, X_b and X_c, which only arises at a subsequent iteration. The initially selected input variable X_a is now redundant, and can be unselected in favour of the pair X_b and X_c.

A common example of this approach is step-wise regression, which is widely used for the development of linear regression models. In this wrapper approach, linear models are iteratively constructed by adding an input variable to the model, and re-estimating the model coefficients. Input variables are retained based on analysis of the coefficients of the newly developed model. The selection process continues until the model satisfies some optimality criterion, such as the AIC (see Section 2.3.1), that is, when $k+1$ input variables are no better than the preceding k variables.

Backward Elimination

Backward elimination is essentially the reverse of the forward selection approach. In this case, all d input variables are initially selected, and then the most unimportant variables are eliminated one-by-one. In wrapper selection strategies, the relative importance of an input variable may be determined by removing an input variable X_i and evaluating the effect on the model that is retrained without it; or, by examining the influence of each of the input variables on the output y through some sensitivity analysis. In filter strategies, the least relevant candidates are iteratively removed until the optimality criterion is satisfied.

A common example of backward elimination is the pruning strategy applied in ANN model development, wherein the connection weights of a network are assessed, and insignificant weights are removed from the network. Pruning algorithms were originally developed to address the computational burden associated with fully connected networks, given that many of the weights may be only marginally important due to redundancy within the ANN architecture. However, the strategy also offers the means of selectively removing inputs by eliminating connection weights between the input and hidden layers.

In general, backward elimination is inefficient in comparison with forward selection, as it can require the development and evaluation of many large ANN models before reaching the optimal model. Since all input variables are initially included, it may be more difficult to determine the relative importance of an individual input variable than in forward selection, which starts with a single input variable. Also, wrapper algorithms based on backward elimination may potentially be biased by overfitting of large models in the same manner as wrappers that utilise global search strategies.

2.3.3 Dimensionality Reduction

The taxonomical classification in Figure 2.1 shows dimensionality reduction algorithms as the first class of algorithms reducing the number of variables within a dataset. Dimensionality reduction is closely related to the task of input variable selection, and is regularly employed as a form of data pre-processing in many multivariate data analysis applications. Dimensionality reduction is performed in order to reduce the computational effort associated with data processing, or to identify a suitable subset of variables to include in the analysis. Comprehensive surveys of dimensionality reduction techniques can be found in *Carreira-Perpinan* (1997) and *Fodor* (2002). However, it is worth highlighting several of these dimensionality reduction techniques, since several hybrid IVS algorithms have been proposed within the literature that make use of them as a pre-processing step ahead of variable selection. The potential benefit is that the identification of a set of informative, yet independent variables, can improve the performance of the input variable selection algorithm.

Principal Component Analysis

Principal component analysis (PCA) is a commonly adopted technique for reducing the dimensionality of a dataset X. PCA achieves dimensionality reduction by expressing the p variables $(x_1, \ldots, x_p)$ as d feature vectors (or, *principal components* (PCs)), where $d < p$. The PCs are a set of orthogonal, linear combinations of the original variables within the dataset. Essentially, PCA can be considered a data pre-processing algorithm that determines an optimal rotational transformation of the dataset, X, that maximises the amount of variance of the output Y that is explained by the PCs (*Fodor*, 2002).

Considering a given dataset X, PCA is performed as follows:

i. Subtract the mean value of each variable, to ensure that $\bar{x_i} = 0$ for each $x_i \in X$.

ii. Find the covariance matrix $\Sigma = \text{Cov}(X) = X^T X$.

iii. Determine the unit eigenvectors $e_1, \ldots, e_p$ of Σ.

iv. Determine the corresponding eigenvalues $\lambda_1, \ldots, \lambda_p$.

v. Rank the eigenvectors according to their eigenvalues.

vi. Select the d PCs according to their eigenvalues.

Selection of PCs is based on examining the eigenvalues of each PC, which correspond to the amount of variance explained by each PC, and thereby including only the significant PCs as input features. A common selection method is to rank the PCs and select all PCs whose eigenvalues exceed some threshold λ_0, or generate a plot of the cumulative eigenvalue as a function of the number of PCs, k, to ensure the selected components explain the desired amount of variance of Y. Another technique is to use and generate a *scree* plot of the percentage contribution of each $k^t h$ PC and to visually identify an optimal value of k (*Fodor*, 2002).

PC regression is a popular application of PCA, where a linear regression model is developed based on the selected PCs. The reduced dimensionality and orthogonality of the PCs substantially improve model parameter estimation. PCA has also been used as the basis for IVS for the development of ANN models (see, for example, *Olsson et al.* (2004), *Gibbs et al.* (2006), and *Bowden* (2003)). However, the mixing of input variables is assumed to be linear, as is the relationship between principal components and the output. Consequently, the application of PCA in this case is flawed, since it will fail to identify any non-linear relationships within the data. Although non-linear versions of the PCA algorithm exist, the transformations of the data can be highly complex, and interpretation of the PCs is much more difficult.

An additional disadvantage of PCA is that the algorithm identifies only important component vectors, rather than variables. Consequently, although PCA may be useful in removing noise from the data, it is not possible to distinguish the unique contributions of individual variables to the variance in the output.

Independent Component Analysis

Independent component analysis (ICA) seeks to determine a set of d independent component vectors within a dataset X. The approach is conceptually similar to PCA, although it relaxes the orthogonality constraint on component vectors. Furthermore, where PCA determines the optimal transformation of the data by considering covariance and identifying uncorrelated PCs based on covariance, ICA considers statistically independent combinations of variables where the order of the statistic that is used can be arbitrary (*Fodor*, 2002). ICA is therefore not restricted to linear correlations, and is more widely applicable to non-linear datasets (*Back and Trappenberg*, 2001). However, like PCA, ICA cannot discriminate unique variables as predictors, and is restricted to determining independent feature vectors.

Vector Quantization

Vector quantization (VQ) refers to techniques that describe a larger set of n vectors by c codebook, or prototype vectors. VQ is closely associated with data clustering and is more commonly associated with algorithms for data compression, in terms of length n. However, *Bowden* (2003) demonstrates the potential for VQ algorithms to be used as an alternative to PCA for data dimensionality reduction. In this case, the d vectors of the candidate set are represented by prototype vectors. Similar candidate variables will be identified by the formation of groups, which have the closest proximity (defined by some distance measure) to the same prototype vector. Note that the algorithm is *unsupervised* and does not assert which of the groups of variables have a strong correlation with the output, merely that candidate variables are associated in some way with each other.

Bowden (2003) implements VQ using the self-organizing map (SOM) (described in detail in Section 8.3). The advantage of VQ algorithms, such as the SOM, over PCA for dimensionality reduction is that they can identify a reduced set of independent variables, rather than feature vectors. An additional advantage is that VQ algorithms, such as the SOM, are non-linear. However, care should be taken in how non-linearity is defined, since it can be interpreted incorrectly, as in *Bowden et al.* (2005). The SOM is considered a non-linear algorithm due to the topological mapping of multidimensional data into a low dimensional space (*Kohonen*, 1995). Non-linear dimensionality reduction could be claimed by analysing the data within this low dimensional space. However, according to the definition of VQ, it is the distance measure that defines the nature of association between two vectors in the SOM algorithm. In the basic SOM algorithm, the Euclidean distance is used, by which the degree of superposition (overlap) of two vectors defines the notion of similarity, rather than correlation. In fact, two vectors may lie far from each other, but be perfectly correlated (for example, consider two parallel lines separated by a distance, D; or, two perfectly, auto-correlated time-series with a phase delay, d). In this case, variables will be grouped according to superposition, rather than correlation, which suggests that only perfect, positive linear correlation between variables will be identified, which in fact makes the algorithm less useful than PCA. In order to claim non-linear dimensionality reduction, it would be necessary to modify the SOM algorithm, by adopting a non-linear measure of correlation as the distance metric. One suitable metric is mutual information, which has been applied previously to clustering algorithms (*Maier et al.*, 2006).

2.3.4 Wrappers

Wrapper algorithms are the first of the two main classes of variable selection algorithm shown according to Figure 2.1. Wrapper algorithms are the simplest IVS algorithm to formulate. Essentially, the algorithm that results is defined by the choice of the induction algorithm (i.e. model architecture). The efficiency of the algorithm will depend largely on the ability of the model to represent relationships within the data and how quickly trial models can be constructed and evaluated.

Single Variable Regression (SVR)

The notion of ranking individual candidate variables according to correlation can be extended by implementing a wrapper approach in order to relax the assumption of linearity in correlation analysis (*Guyon and Elisseeff*, 2003). In this approach, a single variable regression[1] (SVR) is constructed using each candidate variable, which is then ranked according to the model performance-based optimality criterion, such as the cross-validation error. In comparison to ranking filters, SVR can potentially suffer from overfitting due to the additional flexibility in the regression model.

The GRNN input determination algorithm (GRIDA) (*Bowden et al.*, 2006) is a recent example of an SVR wrapper for input variable ranking, which proceeds as follows:

```
Let $X \rightarrow C$. (Initialisation)
For each $x \in X$,
    Train a GRNN and determine $\mathrm{MSE}_x$.
    For $b = 1$ to $100$, (Bootstrap)
        Randomly shuffle $x \rightarrow \varepsilon$.
        Estimate $\mathrm{MSE}_{\varepsilon,b}$.
    Estimate $\mathrm{MSE}_{\varepsilon}^{(95)}$.
    If $\mathrm{MSE}_x > \mathrm{MSE}_{\varepsilon}^{(95)}$ or $\mathrm{MSE}_x > \Theta$ (Selection),
        Remove $x$ from $X$.
Return $X$.
```

where $\mathrm{MSE}_{\varepsilon}^{(95)}$ is the 95^{th} percentile, and Θ is some threshold value.

Considering each variable in turn, a GRNN is trained, and then the MSE of the model is determined for a set of test data. However, rather than greedy selection of the k best variables, each variable is compared to a bootstrap estimate of a confidence bound for the randomised model error, $\mathrm{MSE}_{\varepsilon}^{(95)}$. A variable is rejected immediately if the model error exceeds the randomised error, since it is no better predictor than a random noise variable. Further strictness on selections is imposed through the heuristic error threshold, Θ. However, a suitable value for Θ needs to be determined first. The number of variables selected for a given value of Θ will be dependent on several factors, including the degree of noise in the data, the error function used, and the distribution of the error over the candidate variables. Conseqently, optimal values for Θ can only be determined for each dataset by trial and error.

[1] The term has been adapted from the term single variable classifier (SVC), which is more often referred to within literature due to its application in classification

The estimation of the confidence bound on the error for each SVR is a significant computational requirement. Given the assumed constraint $0 < \Theta < \text{MSE}_{\varepsilon}^{(95)}$, the estimation of the bootstrap may not even be necessary to perform IVS. However, the method does provide useful information in discriminating noise variables from weakly informative ones.

Like all MR filters, the SVR approach does not account for interractions between variables. In order to overcome this, *Bowden et al.* (2006) utilise SOM-based dimensionality reduction, in order to obtain an independent set of candidate variables, prior to selection. However, as discussed previously in Section 2.3.3, the application of the SOM can potentially result in unexpected results.

GRNN Wrappers

Bowden et al. (2005) utilised an evolutionary wrapper strategy for IVS that combined a GA optimisation with a GRNN architecture. The method exploits the fast GRNN training times, and the fixed architecture of the GRNN, which avoids the need to optimise the internal architecture and training algorithm parameters. These are required for the development of other architectures, such as the MLP. A simple binary GA (a GA with decisions encoded as 1 or 0 within a binary string) was utilised, with the objective of minimising the MSE obtained by hold-out validation on a set of test data. In order to overcome the inability of the wrapper methodology to detect interractions between candidate variables, as with GRIDA, *Bowden et al.* (2005) adopted SOM-based dimensionality reduction as a pre-processing stage to reduce the candidate variables to a set of independent variables.

The Chlorcast©methodology (*Serodes et al.*, 2001) for ANN development also utilises a GRNN wrapper approach to optimise the input variables for a GRNN. However, the search space is restricted to nested subsets formed by increasing the order for all candidate variables and an exhaustive search is undertaken to determine the optimal model order, d, for a time-series model. The Chlorcast©methodology does not consider differences in persistence, differences in delays with respect to the model output variable, and redundancies that might be observed within the candidates. Consequently, the method is likely to yield a sub-optimal set of input variables in comparison to the GA wrapper, which searches through many more combinations of variables. Model performance is gauged only on the MSE of the trained GRNN, and although *Serodes et al.* (2001) conclude that the increase in performance of the model for increasing model order was due to capturing more of the process dynamics, it could also be concluded that the models were increasingly over-fitted.

2.3.5 Filters

In the taxonomy shown in Figure 2.1, filter algorithms represent the second subclass of variable selection algorithms and represent an alternative to the wrapper approach. The design of filter algorithms is typically defined by the measure of relevance that is used to distinguish the important input variables, as well as the optimality criteria, as they have been previously defined for filters in Section 2.3.1. Incremental search strategies tend to dominate filter designs, since the relevance measure is usually a bivariate statistic, which necessitates evaluating each candidate-output relationship. Currently, two broad classes of filters have been considered: those based on linear correlation, and those based on information theoretic measures, such as mutual information.

Rank Correlation

Arguably the most commonly used relevance measure in multivariate statistics is the Pearson correlation. The Pearson correlation (also called *linear correlation*, or *cross-correlation*), R, is defined by

$$R_{XY} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.7}$$

where R_{XY} is the short-hand notation for $R(X,Y)$. In (2.7), the numerator is simply the sample covariance, $Var(X,Y)$; and the two terms in the denominator are the square-root of the sample variances, $Var(X)$ and $Var(Y)$. The application of correlation analysis to variable selection originates from linear regression analysis. The squared correlation, R_{XY}^2, is the coefficient of determination, and if X and Y have been standardised to have a zero mean, R^2 is the equivalent to the coefficient of a linear fit between X and Y.

Input variable ranking based on the Pearson correlation is one of the most widely used IVS methods. The selection of candidate variables that are sorted by order of decreasing correlation is based either on greedy selection of the first k variables, or upon all variables for which the correlation is significantly different from zero. The significance of the Pearson correlation can be determined directly, since the error associated with estimation of correlation from a sample is defined by the

t-distribution. A rule of thumb (for large n) is that variables with an absolute correlation greater than $2/\sqrt{n}$ are significant.

Identification of significant correlations is a common technique in data mining applications, such as gene expression analysis, where the goal is simply to mark potentially important genes for further investigation. However, in terms of IVS algorithms, the method is classed as an MR filter, and does not consider interactions between variables. Redundancy is particularly problematic for multivariate time-series forecasting, which considers lagged values that are often highly correlated (auto-correlated).

Partial Correlation

In the case where candidate variables are themselves correlated, redundancy becomes an issue, and a correlation ranking approach is likely to select too many variables, since many candidates will each provide the same information regarding the output variable. Given three variables X, Y and Z, the partial correlation, $R'(X,Y|Z)$ measures the correlation between X and Y after the relationship betwen Y and Z has been discounted. The partial correlation can be determined from the Pearson correlation using the equation:

$$R_{XY\cdot Z}\frac{R_{XY}-R_{XZ}R_{YZ}}{\sqrt{(1-R^2_{XZ})(1-R^2_{YZ})}} \tag{2.8}$$

where $R_{XY\cdot Z}$ and R_{XY} etc. are the short-hand notation for $R'(X,Y|Z)$ and R_{XY} etc.

Partial correlation is similar to stepwise multiple linear regression. The subtle difference is that in stepwise MLR, successive models are fitted with additional input variables, and variables are selected (or later rejected) based on the estimated model coefficients. However, in partial correlation analysis, the magnitude of R' for each variable is not necessarily equal to the regression coefficients for a fitted MLR model, since redundancy between variables means that the solution to the MLR parameter estimation is a line (two redundant coefficients) or a surface, that is, there will be infinite combinations of equivalent model coefficients. The partial correlations obtained are in fact one specific solution to the MLR parameter estimation. Another difference is that forward selection is used in partial correlation analysis, because once the most salient variable has been selected, it will not be rejected later, and the partial correlations of subsequent variables will be dependent on those already selected.

Box-Jenkins

Box-Jenkins time-series analysis (*Box and Jenkins*, 1976), which considers the development of linear auto-regressive, moving-average (ARMA) models to represent dynamic processes, is the most common approach to the development of time-series and process transfer functions. ARMA models are described by the general form

$$y(t+1) = \sum_{k=0}^{p} \alpha_k y(t-k) + \sum_{k=0}^{q} \beta_k u(t-k) \tag{2.9}$$

where α_k and β_k are coefficients and p and q denote the order of the auto-regressive (AR) and moving-average (MA) components of the model, respectively. Identification of the optimal model parameters p and q forms the goal of Box-Jenkins model identification, and hence variable selection. The autocorrelation function (ACF), $R\left(Y(t-k), Y(t)\right)$, determines q and the partial auto-correlation function (PACF) determines p. The ACF is determined for a given time-series sample by

$$R_k = \frac{\sum_{i=1}^{n-k}(x_i - \bar{x})(x_{i-k} - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{2.10}$$

where R_k is the short-hand notation for the auto-correlation of a time-series with a delay of k. The PACF at a delay of k is denoted by ϕ_{kk}, and is estimated from the ACF based on the following series of equations

$$\phi_{11} = R_1 \tag{2.11}$$

$$\phi_{22} = \frac{R_2 - R_1^2}{1 - R_1^2} \tag{2.12}$$

$$\phi_{kj} = \phi_{k-1,j} - \phi_{kk}\phi_{k-1,k-j}, \text{ for } k \geq 2 \text{ and } j \geq 1, \tag{2.13}$$

$$\phi_{kk} = \frac{R_k - \sum_{j=1}^{k-1}\phi_{k-1,j}R_{k-j}}{1 - \sum_{j=1}^{k-1}\phi_{k-1,j}R_j}, \text{ for } k \geq 3. \tag{2.14}$$

The Box-Jenkins methodology can be used to similarly identify optimal linear autoregressive with exogenous inputs (ARX) models. In this case, the partial cross-correlation is used to identify the relevant lags of the exogenous variables.

Box-Jenkins and partial autocorrelation analysis have been used as the basis for IVS in the development of ANN models. In some examples, ANNs have been developed based on an optimal set determined for an ARX model. The ANNs were found to produce better predictions than the ARX model, and this has often provided the justification for ANN modelling in favour of conventional time-series techniques (*Rodriguez et al.*, 1997). However, although this demonstrated the additional flexibility of ANN architectures to describe more complex behaviour, the ANN developed may not have been optimal, since the selection of inputs was based on the identification of a linear model. It may be the case that variables that are highly informative, but non-linearly correlated with the output variable, will be overlooked and excluded from the ANN model.

Mutual information feature selection (MIFS)

The limitations of linear correlation analysis have created interest in alternative statistical measures of dependence, which are more adept at identifying and quantifying dependence that may be chaotic or non-linear; and which may therefore be more suitable for the development of ANN models. Mutual information (MI) is a measure of dependence that is based on information theory and the notion of entropy *Shannon* (1948), and is determined by the equation

$$I(X;Y) = \iint p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \, dxdy, \tag{2.15}$$

where I denotes the MI between X and Y. Further details on the definition and estimation of MI are provided in Chapter 5. MI measures the quantity of information about a variable Y that is provided by a second variable X. However, it is often convenient to simply regard MI as a more general measure of correlation, since despite originating from information theory, rather than statistics, MI is not entirely unrelated to Pearson correlation. In fact, it can be shown that in the case of noise-free, Gaussian data, MI will be related to linear correlation according the relationship:

$$I(X;Y) = \frac{1}{2} \log \left(1 - R_{XY}^2\right). \tag{2.16}$$

The advantage of MI over linear correlation is that MI is based solely on probability distributions within the data and is therefore an arbitrary measure, which makes no assumption regarding the structure of the dependence between variables. It has also been found to be robust due to its insensitivity to noise and data transformations (*Battiti*, 1994; *Darbellay*, 1999; *Soofi and Retzer*, 2003). Consequently, MI has recently been found to be a more suitable measure of dependence for IVS during ANN development. *Torkkola* (2003) also discusses the merit of analysing MI, given that it provides an approximation to the Bayes error rate. Bayes' theorem, which gives the most general form of statistical inference, is given by

$$p(y|x \in X) = \frac{p(y \in Y)p(x \in X|Y)}{p(x \in X)} \tag{2.17}$$

Bayes' theorem can be used to determine the expectation $E\left(y|x \in X\right)$, assuming the probability distributions are known. MI provides an approximation to the error associated with the Bayes estimate of $E(y|X \in X)$, since it can be shown that the minimum estimation error will be achieved for a maximal value of $I(X;Y)$. Consequently, MI provides a generic estimation of the modellability of an output variable Y, which therefore makes MI an attractive measure of relevance in determining an optimal set of input variables, since we would seek the set of input variables that maximises the JMI, that is, the MI between the output and the input variable set.

The MIFS algorithm is a forward selection filter proposed by *Battiti* (1994) to address shortcomings with algorithms based on linear correlation. Considering the candidate set C and output variable Y, the MIFS algorithm proceeds as follows:

```
  i. Let $X \rightarrow \phi$.
 ii. While $|X| < k$,
iii.    For each $c \in C$,
 iv.       Estimate $I(c,Y|X) = I(c,Y) - \beta \sum_{x \in X} I(c;x)$.
  v.    Find $c_s$ that maximises $I(c,Y|X)$.
 vi.    Move $c_s$ to $X$.
vii. Return $X$.
```

MIFS defines a MR filter and identifies suitable candidates according to the estimated bivariate MI between candidate variables and the output variable. The MI between the most salient candidate c_s and the already selected variables in X is estimated and subtracted from the relevance in order to achieve minimum redundancy. The heuristic weighting β determines the degree of redundancy checking within MIFS. If $\beta = 0$, then MIFS will neglect relationships between candidates and MIFS is reduced to a MI ranking filter. Increasing β increases the influence of candidate redundancy on selections, however if β is too large, then the redundancy is overstated and candidate interactions dominate the selection of variables, rather than the input-output relationships (*Kwak and Choi*, 2002). *Battiti* (1994) recommends that a weighting of 0.5–1.0 is appropriate. A criticism of the forward selection approach is that the JMI of the input variable set must be considered in order to correctly determine the optimality of the input variables (*Bonnlander and Weigend*, 1994). However, in MIFS, the forward selection procedure considers variables individually, and optimality of the JMI is inferred by the mRMR selection. The heuristic redundancy parameter β provides only an approximation to the conditional dependence and does not necessarily relate to the JMI.

Partial Mutual Information

Sharma (2000) proposed an IVS filter that is structured similarly to MIFS, but is based instead upon direct estimation of partial mutual information (PMI). The kernel estimation of PMI makes the algorithm ideally suited for application to datasets of continuous variables, and is therefore applicable to environmental modelling applications. The algorithm has been successfully applied to select predictors for hydrological models (*Sharma*, 2000) and ANN water quality forecasting models (*Bowden et al.*, 2002; *Kingston*, 2006).

The PMI-based filter also incorporates a mechanism for testing the significance of candidate variables, so that the termination point of the algorithm is optimally determined, which is an improvement over the greedy selection of k variables in MIFS. In this case, the termination criterion is based upon the distribution of the error in PMI estimation, which is numerically approximated by a bootstrap approach (*Sharma*, 2000). The significance of the most relevant candidate is determined by direct comparison to the upper confidence bound on the estimation error.

The PMI filter algorithm also has advantages over other MI filter designs, such as MIFS, since it is able to identify redundancy and optimize the JMI indirectly via the estimation of PMI. The optimality of the input variable set is ensured because PMI is directly estimated, and the JMI can be determined as a result of the MI chain-rule decomposition, which is given as (*Cover and Thomas*, 1991)

$$I(x_1, \ldots, x_p; y) = I(x_1; y) + I(x_2; y|x_1) + \cdots + I(x_p; y|x_1, \ldots, x_{p-1}). \tag{2.18}$$

Recall that in MIFS, the JMI cannot be directly approximated because redundancy is only approximated by a heuristic weighting factor. The termination criterion in PMIS automatically determines the optimal number of input variables, since the increase in JMI is additive, and once the contribution of an additional input variable is insignificant, the selection process terminates and the JMI will be maximised.

An additional benefit of PMIS is that the information yield during IVS provides a useful indication of the contribution of each input variable to the prediction of the output variable. Several methods for determining the usefulness of input variables based on analysis of the trained model have been described and range from sensitivity analysis, to aggregation of the weights associated with each input variable. However, the relative importance of an input variable can be determined statistically from the MI between each input and the output variable (*Soofi and Retzer*, 2003). The PMI estimated for a given variable can potentially also be used to classify input variables as informative, or weakly informative, as defined by *Kohavi and John* (1997), by considering the conditional relevance. *Kingston* (2006) considered several techniques for determining the relative importance (RI) of input variables and found that the method based on PMI yielded similar estimates of RI as methods based on analysis of the connection weights for a trained MLP. The method for estimating RI was based on the formula

$$\mathrm{RI}(i) = \frac{I'(x_i; y)}{\sum_{x \in X} I'(x; y)}, \tag{2.19}$$

where I' denotes the PMI estimated for candidate variable x during PMIS.

The usefulness of RI is that it provides an indication of the way in which the ANN generates predictions. Although it is assumed that the ANN is using all of the input variables, it may in fact only require some small subset of the available input variables to generate predictions. In this case, further refinements to the ANN can be made based on this interpretation. Such considerations might be important when considering the cost of data collection that is associated with ongoing deployment of ANN models. One might consider sacrificing model accuracy in favour of cost reductions in ANN maintenance by reducing the number of input variables even further, which would reduce data requirements. The RI of variables may also encourage increased efforts toward the development of measurement techniques to ensure data quality for important variables.

The main limitation of the PMI filter is the computational effort associated with the bootstrap estimation of MI. Even obtaining a single estimate of MI is a naturally expensive computation due to the $O\{n^2\}$ density estimation, and computational efficiency is therefore influenced by the sample size, n. The bootstrap therefore significantly adds to the overall burden by increasing the number of estimations of MI required to implement IVS. *Sharma* (2000) restricts the size of the bootstrap to 100 in order to maintain reasonable analysis times. However, a small bootstrap of this size might compromise the accuracy of the termination criterion, since potentially the confidence bounds may be poorly estimated.

Hybrid ICA and IVS filter (ICAIVS)

A hybrid ICA and IVS filter algorithm (ICAIVS) was proposed by *Back and Trappenberg* (2001), which considers the combined statistical relevance of input variables in deciding whether or not a variable should be included. ICAIVS consists of two main steps: (*Trappenberg et al.*, 2006)

i. ICA: Produce a set of candidates which are as statistically independent as possible.

ii. IVS: Perform a set of statistical tests between the independent candidate variables and the desired output variables.

Here, the statistical analysis is based on estimation of the joint dependence of combinations of input variables and considers all combinations from $c(x_1^p, y)$ through to $c(x_1^p, \dots, x_n^p, y)$, where p denotes the order of the dependence that is measured. The IVS procedure then compares the relevance for each subset of variables, with respect to the average dependence for all subsets, and a subset is selected if the dependence exceeds some threshold value K.

The drawback of ICAIVS is that the algorithm does not scale well, given the large number $(3^n - 1)$ of statistical tests that must be performed, considering only second-order statistics. Recently, an improved version of ICAIVS was described that utilised MI as the statistical measure of dependence (*Trappenberg* *et al.*, 2006). This reduced the number of statistical tests by considering only first order MI. However, the problem of specifying a suitable threshold value still remains. Both *Back and Trappenberg* (2001) and *Trappenberg et al.* (2006) used a value of 0.2 for this threshold. However, in this case the threshold value is heuristically determined and a suitable value may vary depending on the dataset (pers. comm. A. Back, 2007).

2.4 Comparison of Approaches

The benefits and limitations of various IVS strategies or algorithms are summarised in Table 2.2. Here, the different algorithms are evaluated and compared according to a number of criteria. Firstly, whether the algorithm is suitable for identifying non-linear relationships—a fundamental requirement for the development of ANN models. The choice between model-free and model-based IVS algorithms may also be a consideration, given the restriction imposed by wrapper designs on the choice of model architecture. The computational efficiency and scalability of the algorithm are also important, in particular where there are computational constraints due to available hardware. Finally, the optimality of the selected variables and the degree of redundancy checking represent the quality of the solution that is obtained by the algorithm is also important. Recall from Section 2.2 that the goal of IVS is to achieve the best possible subset of input variables, with minimum redundancy. Furthermore, models with increased redundancy are likely to be more difficult to interpret.

Analysis of linear dependence forms the basis of many ranking schemes, linear model identification, and PCA—which have all been previously applied to the development of ANN models of environmental processes. In these IVS algorithm designs, the correlation, R, is the adopted measure of dependence between variables. The correlation coefficient is very straightforward and fast to compute. PCA is the least scalable of the linear algorithms in Table 2.2, due to the d^2 computation of the covariance matrix, Σ. Otherwise, the linear IVS algorithms are highly efficient, and can be used to determine the saliency of large numbers of candidate variables. Box–Jenkins or partial correlation would be used in preference to correlation ranking schemes because of the improved quality of the input variable set that is achieved by handling redundancy through the estimation of R'. However, key issues relating to linear IVS filters are the sensitivity of the linear correlation coefficient to noise, and to data transformations during preprocessing, which can influence the apparent relevancy of input variables (*Battiti*, 1994). Most important, however, is the questionable suitability of these methods for ANN development, since the underlying assumption of linearly structured dependence is contradictory to the development of statistical models of non-linear systems.

Table 2.2 compares four IVS wrapper designs (GA, GRIDA, forward and backward selection), which are representative of the many different wrappers that can be formed using different combinations of ANN models, search methods and optimality criteria. GRIDA is a relatively efficient wrapper design that exploits the speed of GRNN development, however the error-based termination criterion is somewhat arbitrary, which makes it difficult to apply. The GA-based approach is arguably the most promising of the IVS wrapper approaches, in comparison to other standard wrapper designs such as forward selection and backward elimination. It provides an efficient means of automating the trial-and-error evaluation of different input variable sets, since the GA can efficiently search the potentially large number of combinations of input variable sets to determine a near-globally optimum set. However, the approach is subject to scalability constraints. Here, the scalability refers to the growth in the search space, which is exponential with respect to the dimensionality of the candidate set, which may affect the number of solutions that must be evaluated to have confidence that the final solution is the global, or at least near global, optimum.

Regardless of the implementation, the potentially high computational expense of the trial calibration and evaluation of a large number of models is considered to be the predominant weakness of using any IVS wrapper (*Kwak and Choi*, 2002; *Chow and Huang*, 2005). Furthermore, optimal performance of the trained ANN does not strictly imply optimality of the input set, since this is also dependent on additional factors such as the type of ANN architecture, training algorithm, and the optimality criteria adopted. The appropriateness of a set of inputs obtained for a particular model architecture is therefore not guaranteed for another, and restricts the applicability of any input set obtained using a wrapper technique (*Battiti*, 1994). Furthermore, wrapper strategies are essentially holistic in their evaluation of the input variables, since they only consider the performance of the network trained with the variables. It is difficult to determine the effect of an individual input variable on the output, especially when there is redundancy within the candidate data, which is considered to be another disadvantage of this approach.

In contrast to the *model-based* wrapper approach, *model–free* filter techniques utilise a statistical measure of the degree of dependence between the candidates and output variables as the basis for input variable selection. The separation of the IVS task from the model calibration and selection tasks not only yields a more efficient algorithm overall, but the resulting input set has wider applicability to different model architectures (*Kohavi and John*, 1997; *Guyon and Elisseeff*, 2003). However, the performance of IVS filters is largely dependent on selection of a suitable statistical dependency measure for the application at hand.

Information theoretic dependency measures, such as MI, offer a highly suitable measure of relevance for IVS filter designs that can be applied successfully to ANN development. In particular, the underlying generality of the measure of dependence provides a sound basis for model-free estimation of the relevance of input variables. Here, four filter designs are compared: MIFS, PMIS, JMI and ICAIVS, which are all based on the estimation of MI. However, several issues have arisen in the formulation of MI-based IVS algorithms, which are: the additional computational effort in estimating MI, the ability of the selection algorithm to consider the inter-dependencies between candidates (i.e. redundancy checking); and the lack of an appropriate analytical method for determining when the optimal set has been selected (*Chow and Huang*, 2005). The IVS filter design proposed by *Sharma* (2000) overcomes several of these difficulties using the concept of PMI. The PMIS algorithm is a relatively efficiently structured forward selection algorithm, and the usefulness of PMI to provide a model-free measure of the relative importance of input variables is an added advantage of this approach, as the selected input variables can be analysed to determine the important relationships within a given process. ICAIVS is similarly a suitable approach, but is less preferable due to the high computational requirement, the need for ICA pre-processing, and poor scalability.

On the basis of this review and evaluation of IVS strategies, it is proposed that further investigation of the PMIS filter design is warranted. Although it is a relatively new approach, several studies have found that the application of this algorithm to ANN development in environmental modelling applications has significant merit (*Bowden et al.*, 2005). In comparison to all other IVS strategies, the PMIS method compares favourably in all aspects, with the possible exception of computational efficiency. In this resepect, the main drawback of PMIS is related to the expense of PMI estimation and accurately determining the optimum number of input variables. However, it is likely that further improvements to the existing algorithm, such as the use of the average-shifted histogram (ASH) for density estimation (*Fernando et al.*, 2009), can yield further reductions in the computational requirement, which will add increased efficiency to an already flexible and informative IVS design.

Table 2.2: Comparison of Various IVS algorithms

Algorithm	Suitability				Computation		Solution Quality	
	Criteria	Non-linear	Model-free	Selects Variables	Efficiency	Scalability[a]	Optimum	Redundancy Checking
PCA	Covariance	No	Yes	No	Excellent	Squared	Global	None
Correlation ranking	Correlation	No	Yes	Yes	Excellent	Linear	Greedy	None
Box-Jenkins	ACF/PACF	No	No	Yes	Excellent	Linear	Local	Good
Partial correlation	Correlation	No	Yes	Yes	Excellent	Linear	Local	Excellent
GA-GRNN	Error	Yes	No	Yes	Good	Exponential	Global	Poor
GRIDA	Error	Yes	No	Yes	V. Good	Linear	Greedy	Poor
Forward selection	Error	Yes	No	Yes	Poor	Linear	Local	Poor
Backward elimination	Error	Yes	No	Yes	Poor	Linear	Local	Poor
MIFS	MI	Yes	Yes	Yes	V. Good	Linear	Greedy	Fair
JMI	MI	Yes	Yes	Yes	Poor	Exponential	Global	Good
PMIS	PMI	Yes	Yes	Yes	Fair	Linear	Local	Excellent
ICAIVS	MI	Yes	Yes	Yes	V. Poor	Exponential	Global	Good

[a] *Growth in computational requirement with increasing number of candidates*

Chapter 3

Data Splitting

"In God we trust, the rest have to bring data."

W. Edwards Deming (1900–1993)

3.1 Introduction

Generalisation is a central issue in defining appropriate methods for the development of all statistical models, including ANNs. Statistical models of all type are invariably developed based on a finite set of training data. It is rare that data collected through observation of a process will be noise-free, and the data available for model development are likely to contain a small proportion of features that are not representative of the underlying system. Generalisation refers to the ability of a statistical model to accurately represent the underlying data generating process, rather than the idiosyncratic features of the training data. The latter phenomenon is referred to as *over-fitting* because it is characterised by a high goodness-of-fit to the training data, yet poor performance when querying previously unseen data. Despite their many advantages over conventional statistical models, artificial neural networks remain susceptible to poor generalisation, which can largely be attributed to the complexity of the model architecture (i.e. the number of estimated parameters) relative to the number of training data.

This chapter addresses the issue of generalisation by considering the sampling methods used for selecting ANN training data. Although various ANN training methods can be used to ensure good generalisation is achieved, the sampling of training data can have a significant effect on the quality of training, and on performance assessment. However, the impact of this sampling task on the quality of ANN developed is rarely appreciated by ANN modellers.

In order to increase understanding surrounding this issue, a review of ANN generalisation methods is briefly presented to highlight the importance of data sampling in implementing generalisation techniques, in particular the highly popular hold-out validation approach. Methods for sampling ANN training data are then reviewed and a stratified sampling design is developed, with significant improvements to an implementation based on the self-organizing map (SOM). The improved stratified design is then compared to alternative sampling methods by comparing results for a number of experimental problems. Although ANN data sampling methods are presented individually in many papers, a comparison between different methods has not been extensively undertaken, and the review and subsequent experimental investigation in this chapter goes some way to providing such an evaluation. Ultimately, the comparison of sampling methods and experimental investigation yields some guidelines for choosing an appropriate technique for generating data samples for ANN training.

3.2 Generalisation and Over-fitting

The focus of this chapter is on poor ANN generalisation in the sense of over-fitting, and the notion that poor use of the available modelling data can lead to poor generalisation, or a biased model due to the phenomenon of "over-fitting". *Sarle* (1997) observes that poor generalisation is also symptomatic of a model with insufficient complexity to describe all behaviour of the data generating process. The reasons for this may be either an over simplistic model architecture (i.e. too few internal parameters), or an insufficiently informative set of model input variables. Methods for input variable selection are discussed in Chapter 2 that can ensure an optimal degree of model complexity with respect to the input variable set.

Figure 3.1 illustrates the concept of over-fitting by considering a simple univariate regression problem. In this case the data generating function is $f(x) = \sqrt{x} + \varepsilon$ where $\varepsilon \sim N(0, 0.01)$ and 50 samples are generated uniformly on the domain [0,3]. As shown in Figure 3.1(a), a model architecture with many parameters potentially can fit not just the underlying $\sqrt{x}$ process, but also will fit the noise in the sample of training data. Consequently, the error of estimates from the true process – the validation error – is expected to be high, and the model is said to have poor generalisation performance. A model with fewer parameters is shown in Figure 3.1(b). The degree of over-fitting in this case is reduced, although there remains some influence of the model and it is slightly over-fit. A generalised fit is shown in Figure 3.1(c), in which the fitted model has sufficient complexity to represent the data generating process without over-fitting. Figure 3.1(d) illustrates an under-fit model, which has insufficient complexity to wholly describe the relationship within the data.

An over-fit model has a low error, or *bias*, but the error achieved will be highly dependent on the data, that is, the model error has high *variance*. The opposite is true for the generalised model, which has a higher bias, but which will be less sensitive to the data and will therefore have reduced variance. Ideally, the best model would have both a low error and low variance, but usually for statistical models based on a finite sample of noisy data, this is not possible. Instead, model development is required to trade-off the relative amount of bias and variance, and this is referred to as the *bias-variance dilemma* (*Geman et al.*, 1992). Generalisation for ANN models built on noisy data typically represents a trade-off in which the finite-sample variance is lowered by allowing for a bias that reflects the error due to the naturally occurring noise in the data.

A number of alternative techniques can be adopted during model calibration to ensure that the calibrated model is able to generalise, and does not over-fit the training data. The methods fall into the broad categories:

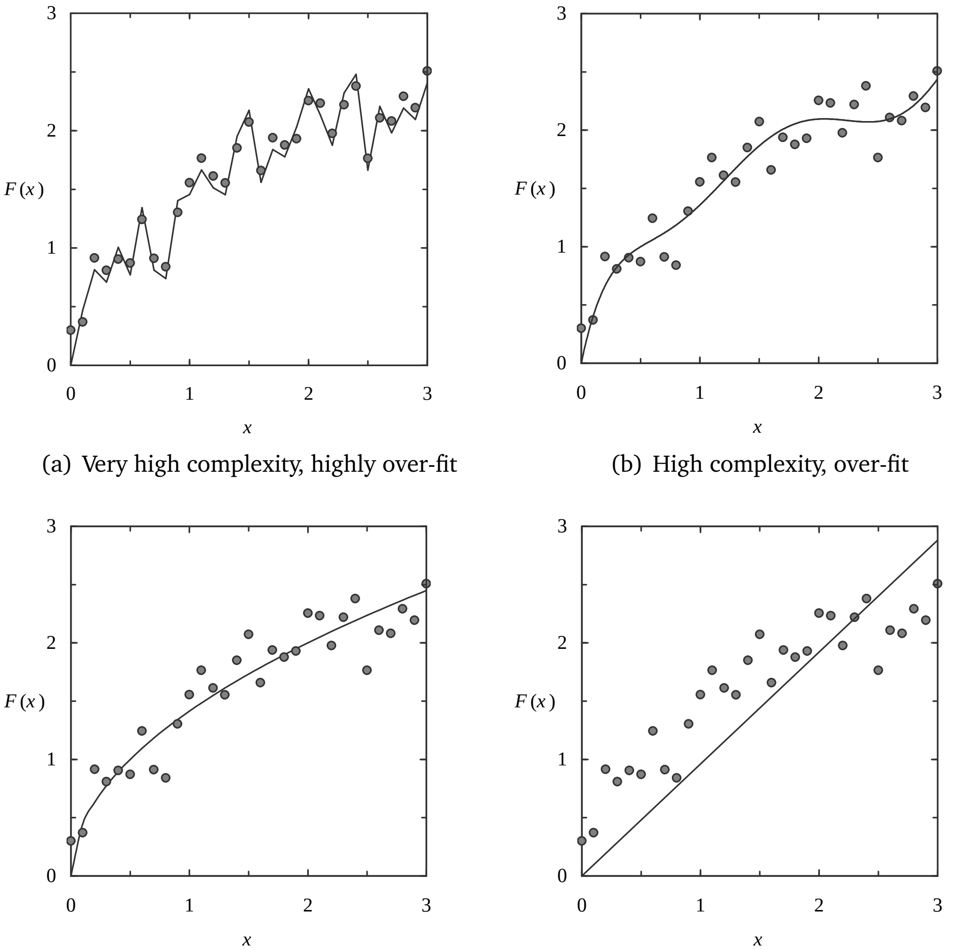
- cross-validation,
- ensemble training, and
- regularisation.

3.2.1 Cross-validation

In general, cross-validation refers to techniques in which one portion of the available data is used to estimate model parameters, and the remaining data are used to independently test the generalisation performance of the trained model. Provided that the training and test data are equally representative of the modelling domain, cross-validation can ensure that over-fitting is avoided.

Hold-out cross validation

In machine learning, the hold-out method of cross-validation is commonly associated with stop-training (or early-stopping). Given a sufficient number of connection weights (i.e. internal parameters) and sufficient training time, an ANN can represent exactly the data within the training set. However, this is not desirable and consequently, hold-out validation is commonly employed in the form of



(a) Very high complexity, highly over-fit

(b) High complexity, over-fit

(c) Sufficient complexity, generalised fit

(d) Low complexity, under-fit

Figure 3.1: Phenomenon of over-fitting. An overly complex model (a) will have a small error (low bias), but will be highly dependent on the sample (high variance). The degree of overfitting decreases for fewer parameters (b) until the most generalised model (c) is determined. Insufficient complexity (d) can underfit the data.

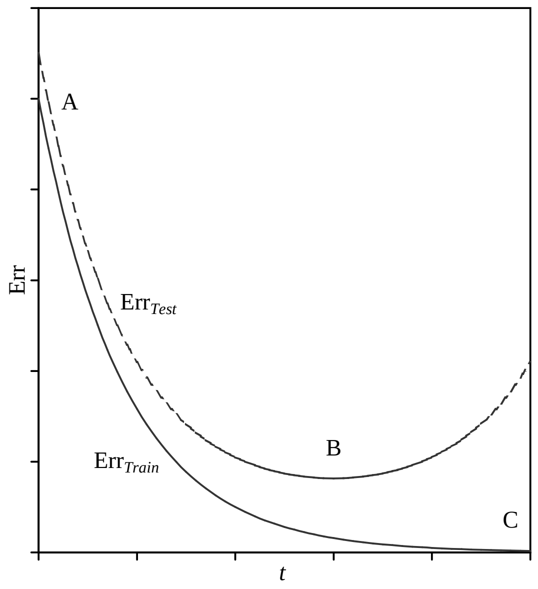


Figure 3.2: Stop training (early-stopping) using test data to ensure generalisation during ANN training.

stop-training. In this approach, the training data are used to guide the learning process and the test error is periodically determined to ensure that the model remains general. As shown in Figure 3.2, at A the initial error for both training and test data will be poor for a randomly initialised ANN. During training, the error reduces as the ANN learns the relationships within the data until, at some point (B), the optimal generalisation performance is achieved. Further training will reduce the training error of the network, but the test error will not improve and may in fact increase as the ANN begins to over-fit the training data. Given sufficient time, the ANN will be trained to perfectly represent the cases within the training data (C), but will poorly represent the underlying process. The training is therefore stopped at B, and hence the method is referred to as stop-training or early-stopping.

ANN development requires that two hold-out datasets are generated for testing and validation. The test data are used to implement hold-out validation to avoid over-fitting. However, because the minimisation of the test error is used to determine the optimal training and model parameters, the trained model is said to be optimistically biased towards the test data, that is, the error for the test data may in fact be better than the true validation error. Consequently, it is necessary to undertake an additional validation of the final ANN model, to ensure that true generalisation has been achieved, and hence validate the ANN model (*Maier and Dandy*, 2000).

k-fold cross-validation

In k-fold cross-validation, the data are sampled into k subsets of equal size. The ANN is trained k times using each k^{th} subset as the test data, and the remaining data are used for training. After training has been repeated for all k cases, ANN model performance is estimated by the average for all tests, and an aggregate set of parameters is determined in order to construct the final ANN model. The benefit of the k-fold approach is that all of the data are used as training data, which eliminates the potential for hold-out bias since all available information is utilised. In particular, k-fold is considered a highly suitable cross-validation technique for small datasets, or where training data is sparse.

The choice of k can affect the performance of the technique in terms of both statistical properties of the model error, and the computational effort required. Typically $k = 10$ is considered to be a suitable choice, representing a trade-off between improved model performance and the number of training sets used. The special case of $k = n$ is more commonly known as leave-one-out cross-validation and is also a common choice for small datasets.

3.2.2 Ensemble Training

Ensemble training techniques refer to methods that involve the training of a collection of ANN models, rather than a single model. Each individual component ANN model is trained on a sample of the available training data. These are then later combined or aggregated to give an overall model prediction. The two most common of these approaches are bagging and boosting (*Zhou et al.*, 2002; *Anctil and Lauzon*, 2004). Bagging refers to bootstrap random sampling (with replacement) of multiple training data sets that are each used to train a component network. Boosting aims to improve the performance of ANNs by resampling the training data with increased weighting given to data that correspond to a high prediction error. Initially a random sample is drawn and used to train the first component model and then each subsequent component network in the ensemble is trained on data that are sampled according to the error of the previous model. The aggregation of ensembles in both bagging and boosting typically uses the mean prediction for *all* models, although some methods for determining an optimal subset of the models have also been described (*Zhou et al.*, 2002).

Ensemble techniques generally extend existing cross-validation methods, and can be applied to either $k-$fold or hold-out validation. The benefit of ensemble techniques is that all of the available data are used at some stage during training, and that no data are omitted. Resampling effectively increases the statistical efficiency of ANN learning, since during cross-validation there is a loss of information due to the hold-out portion of data that is never presented during training. Secondly, since multiple training sets are considered, any bias and variance introduced for each instance of a training set is aggregated over all cases, and consequently resampling methods are able to yield ANN models with higher precision, and less variability. Another advantage of ensemble techniques is that the bootstrap training of multiple ANN models allows computation of confidence bounds on predictions.

The main limitation of ensemble techniques is, of course, the computational effort required to train the ensemble of ANN models. The computational issues and considerations are similar to those of wrapper strategies for input variable selection (discussed in Section 2.3), especially for complex ANN architectures where training an individual model may take some time. It is also unclear how an ANN model built using ensemble methods is to be validated, unless there is an initial portion of data removed that is not used in ensemble training. If this were not the case, there is a chance that the training of the ANN model will include data that are later used to validate the performance of the model.

3.2.3 Regularisation

Bayesian regularisation or weight decay aims to minimise the magnitude of connection weights within a neural network. Since large weights often correspond to erratic changes in output values for small changes in the input variables, keeping the connection weights small yields a smoother response, and therefore reduces the variance of the model output. In weight decay, the decrease in value of the weights over successive learning iterations offsets large increases due to the training error. In Bayesian regularisation, the training minimises a modified cost function that includes both the mean prediction error and the squared-sum of the connection weights, such as

$$E = \sum_{w \in W} w^2 + \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{3.1}$$

where w is an individual connection weight, W denotes the set of all connection weights in the ANN model, and second term on the right-hand side is simply the mean squared-error (MSE).

3.3 The Hold-out Bias and Variance Dilemma

Hold-out validation is by far the most common approach to ANN training, as it offers the simplest approach for ensuring that generalisation is achieved when developing ANN models. Recall that the motivation behind hold-out validation is to achieve good generalisation in ANN modelling by addressing the traditional bias and variance dilemma. The magnitude and variability of error due to the estimation of parameters can be optimally balanced by selecting representative training and test data and implementing hold-out validation. However, in implementing hold-out validation, the dilemma is that the hold-out itself may prove to be another possible source of bias and variance. If data are selected inappropriately, then the training, test and validation data may not be equally representative of the problem domain, and this will be manifest as bias in the test and validation errors; or, the results may be sensitive to the specific data used, in which case the test and validation results will be highly variable and will lower the degree of confidence in the developed model. The hold-out is a particular challenge when data are sparse, and hence the growing interest in techniques such as k-fold cross-validation, ensemble ANNs, or regularisation. However, none of these methods can truly avoid the need for at least one hold-out sample to perform a validation of the ANN model. Cross-validation techniques and ensemble techniques should still use hold-out data to validate the performance of the final ANN model that is developed, and regularisation addresses the variance of a model, but not the bias due to the training sample. Consequently, all ANN model development methods will be influenced to some degree by the sampling of data, regardless of the approach to training that is adopted. It is therefore important that the issues surrounding the hold-out bias and variance dilemma are addressed during model development—particularly when implementing a conventional hold-out test and validation approach, since the test and validation samples are generated only once.

In order to overcome potential bias and variance, the two issues to address when implementing the hold-out approach are:

- the proportion of the data in each data subset, and
- how to best allocate the data into subsets.

The relative proportion of each set of data will influence how much data the ANN can utilise during training to reinforce the relationships within the data, and how much variance could be expected due to the specific data in the test and validation sets. Bias and variance analysis has been undertaken for many ANN validation techniques to assess the impact that the proportion of data has on the variance and bias of cross-validation (*Twomey and Smith*, 1998). Too few data in either set can bias the training process, or the performance assessment, towards the particular data in the respective data set. Furthermore, since greater weight is given to each particular observation within in a smaller set, the approach will be more sensitive to the specific data that are used, and the variability of the results will therefore be greater.

In general, a hold-out proportion of 20–50% is used for allocating data (*Bowden et al.*, 2002). The simplest form of allocation is to divide the data evenly between all subsets. If an even proportion (50% hold-out for training and testing, or $1/3$ each for training, testing and validation) is allocated, there is less likely to be bias either way. However, an ANN can benefit from additional information during training, and more data can be allocated to training to improve learning, provided that this can be done without comprising the test or validation sets. The additional data in the training set provide extra examples that can reinforce the underlying relationships, which may help in the case when data are noisy, or the relationship being modelled is highly complex. In most applications that allocate more data to training, the proportion of training data is only slightly above 50% of all available data, with the remainder divided evenly between test and validation data. For example, *Baxter et al.* (2001) suggest that 60% of data are allocated to training, with 20% each allocated as test and training data. *Bowden et al.* (2002) recommend that a hold-out of 20% for validation, and a further hold-out of 20% of the remaining data (16% overall) be used as test data.

The second issue of how to allocate data into subsets is arguably more important, since even though appropriate proportions of data might be used, the respective samples for training, testing and validation might be allocated inappropriately. The most important aspect in this regard, is that of representativeness of the data, in that each of the subsets contains examples of the entire modelling domain. In particular, training is likely to be less useful if the training data do not contain the necessary examples to describe all input-output relationships, and the ANN will be required to extrapolate—something that ANNs do not do very well (*Sarle*, 1997). A related concern is the bias of a model due to the relative frequency of different conditions that occur within the training data. Sparse data (i.e. less frequent training examples) will have less of an influence on ANN training than samples that occur more frequently. The result is that the ANN will learn to predict the majority of cases accurately, but will not perform as well in rarer instances, and the model is considered to be biased towards the more "average" conditions in the training data. However, it is often the case that the less frequent examples are of equal, if not greater, importance and such poor predictions are therefore unacceptable. Such issues are often encountered in the development of ANN classifiers in medical diagnosis applications (*Tourassi and Floyd*, 1997), although this has relevance to environmental applications dealing with ANN-based prediction of rare or infrequent events, such as algal blooms and floods. Finally, it is important that all subsets for training, testing and validation are equally representative of the modelling domain so that there is no bias in the assessment of model performance (*Maier and Dandy*, 2000).

On the subject of selecting training data for ANN development, *Sarle* (1997) simply notes that

> "Methods for selecting training data can be found in statistical textbooks."

However, ANN modellers are rarely statisticians, and despite the strong similarities between ANN and conventional forms of statistical regression, the methods employed during ANN development are rarely considered with the same rigour. The selection of training data is no exception. In many applications, the sampling of data for ANN hold-out validation has been at best random or judgemental, with a general disregard for the effect of unrepresentative samples on training and performance assessment (*Maier and Dandy*, 2000).

Sampling theory is a branch of applied statistics, which considers the effect of sampling on the performance of statistical estimators and regression. Analysis of the dependence of a statistical estimate on the sample can ensure that suitable sampling techniques are devised to yield optimal estimators. However, although sampling methods for survey design and analysis are well-established in this area, the application of similar theory to ANN training data selection is less prevalent. The main limitation is that ANN models are a form of non-parametric regression, and therefore the analysis of sample effects on ANN estimates is not always as straightforward as for conventional statistics, such as the mean and variance. Perhaps another more practical limitation is that many of the available software packages for ANN development do not currently implement many of the sampling algorithms. One plausible reason for this is that a major focus has been on ANN architectures and learning algorithms, and that it is often assumed that the necessary training, test and validation data are available; certainly, this is evident in view of the statement given by *Sarle* (1997).

The purpose of this chapter is to review the hold-out bias variance dilemma by considering the methods employed for the selection of training data within the context of data sampling. In particular, this chapter considers the following questions:

1. What sampling techniques could be used for data splitting?

2. Which sampling methods are most suitable for ANN training data selection?

3. How should data splitting be implemented to achieve the best quality sample (hence, best quality ANN training)?

Note that the definition of the holdout bias dilemma given here immediately implies the notion of sample *quality*, since a desirable sampling method will achieve both low bias and low variance. In other words, the technique used to generate ANN training data should consistently select a sample that results in a highly accurate, generalised model. It is also possible that the sample *efficiency* could be improved by careful selection of training data, such that the information in the training data is maximised with the fewest number of examples, which would reduce the computational requirement of training.

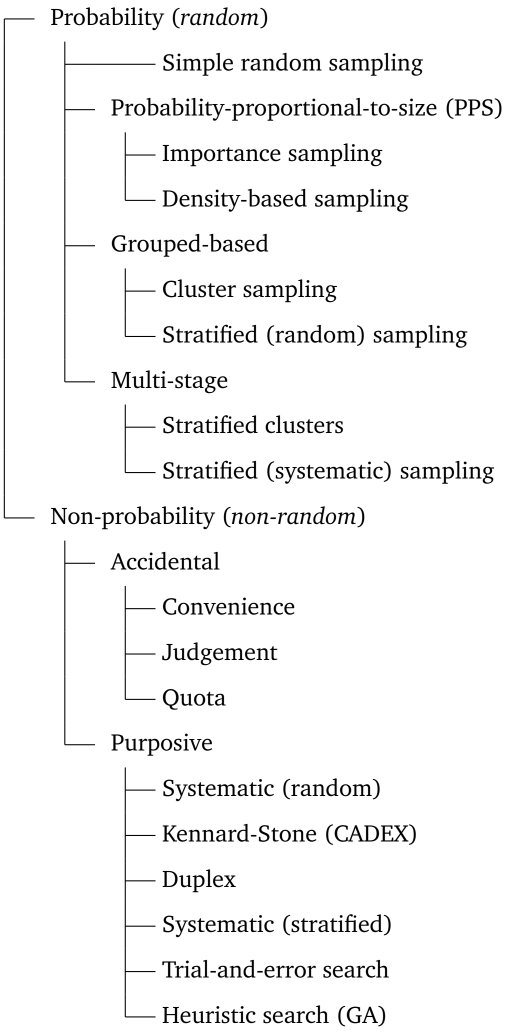
3.4 Sampling Techniques

In the following section, sampling methods for the selection of training, test and validation subsets required for ANN model development are reviewed. Figure 3.3 presents a taxonomy of the various techniques that have been described for data sampling, either for general sampling and survey design, or specifically for the selection of ANN training data. Sampling methods fall into one of two broad categories: probability sampling (random), or non-probability (deterministic) sampling (*Cochran*, 1977).

3.4.1 Probability Sampling

Probability sampling includes sampling methods where each sample is selected with a known probability. In general, probability sampling can select a sample with reduced bias, however, the randomness does create variability of the sample taken. A feature of these sampling methods is that the probability of sample selection can be calculated for each unit within the available data, which allows inference of the potential sample bias and variance of estimates. Hence, for probabilistic sampling, the quality of the sampling method can be determined.

Figure 3.3: Taxonomy of sampling methods for the selection of ANN training data.



Simple Random Sampling (SRS)

Simple random sampling (SRS) is the most basic form of sampling. Given a set of N data, the data are drawn with uniform probability n/N, where n is the total number of samples drawn. Computationally, SRS can be extremely efficient to implement, and some algorithms exist that can generate samples with a single pass over the data (*Knuth*, 1997).

SRS is the most common random sampling technique used in ANN development. In addition to the efficiency and simplicity of the method, SRS can generate an unbiased sample, since the chance of taking any point is equal. Yet, as many researchers are becoming aware, SRS can often result in a poor sample. The pure randomness of the sampling technique also results in a chance that the datasets for training, testing and validation are not representative of each other due to the chance allocation of data amongst the samples. Furthermore, the sampling method is actually naturally biased, by virtue of the probability distribution of the data, towards data with a higher probability i.e. dense regions. Consequently, the sampling may exclude important patterns that occur with less than average frequency, which may impact on model performance.

Importance Sampling

One of two common probability-proportional-to-size (PPS) sampling techniques is importance sampling. In this case, the probability of sampling is determined by the notion of the importance of a given unit, with samples selected with probability proportional to importance. The limitation of importance sampling is that the importance must be evaluated, which requires some knowledge of the data in order to perform the sampling.

Dynamic subset selection (DSS) is a training technique that adopts importance sampling to reduce the computational load associated with supervised learning algorithms (*Gathercole and Ross*, 1994). In DSS, the prediction error for each training observation provides the importance weighting. It is considered more efficient for the learning algorithm to focus on examples that are contributing the most to the overall prediction error. At each epoch during training, the training data are ranked in order of importance. The importance is higher for cases where the model gives a high error. The next phase of training considers only a sample of the most important observations. Periodically, the errors of all training data are determined to re-evaluate the importance rankings for all data to generate subsets for subsequent training iterations.

Density Biased Sampling (DBS)

Density biased sampling (DBS) (*Kollios et al.*, 2003; *Palmer and Faloutsos*, 2000; *Nanopoulos et al.*, 2002) is a form of PPS sampling where the probability of selection is biased according to the local density of the data surrounding each sampling unit. DBS is a generalised form of uniform sampling, and can also be considered a form of importance sampling, where the density of the data defines the notion of importance. Given an estimator for the density function f for X, DBS samples each point with probability p_s defined by

$$p_s(x \in X) = \frac{n}{N}(f(x \in X))^a \tag{3.2}$$

where n is the desired sample size, N is the dataset size, and a is the density bias. The bias controls the sampling, and the following biases can be considered: (*Kollios et al.*, 2003)

- $a = 0$. DBS reduces to SRS, since data are sampled with uniform probability n/N.
- $a > 0$. Over-samples regions of high density, since $p_s(x) > p_s(y)$ if and only if $f(x) > f(y)$, and less dense regions are under-sampled. Sampling probability will be greater than uniform sampling for $f(x) > \bar{f}(x)$. Noise and outliers can be effectively ignored by applying a positive bias.
- $a < 0$. The exact opposite of $a > 0$, the bias increases the sampling rate of sparse regions. A bias on $[-1, 0)$ allows increased sampling of sparse data.

Essentially, the bias function is equivalent to a transformation of the distribution of the data prior to sampling. *Kollios et al.* (2003) proves that provided that $a > -1$, the sampling rate can be adjusted as desired while preserving the relative distribution of the data. Hence, high density data within X remain proportionally dense within the sample. The adjustable bias in DBS offers greater flexibility, since the sampling can be tuned depending on the needs of the application at hand (*Kollios et al.*, 2003).

DBS is considered to be superior to simple random sampling in applications where the distribution of data is non-uniform, which is likely to be the case in most real-world datasets (*Palmer and Faloutsos*, 2000). DBS has been found to improve the performance of ANN classifiers in medical diagnosis, where the occurrence of interesting cases may only be few. The biased sampling was found to add weight to these cases during training to ensure that the ANN was able to fit these data with greater accuracy (*Tourassi and Floyd*, 1997). In hydrological applications, the same considerations could be applied to rainfall events, which are infrequent and of relatively short duration. In water quality datasets, excursions outside of compliance targets are typically rare, however, obtaining accurate predictions for these events is paramount. In each of these cases, training data quality may be improved by implementing DBS to increase the proportion of interesting data.

Like importance sampling, the main limitation of DBS is that it requires that the density of the data is either known exactly, or can be approximated efficiently. *Kollios et al.* (2003) utilise a kernel density estimation (KDE) approach that separates density estimation and sampling, which allows for flexibility and simple implementation. However, the KDE approach may be inefficient for high dimensions, or very slow for lengthy datasets. In the case of the latter, some speed up can be achieved by estimating the density on a random sample of the data (*Kollios et al.*, 2003). A more computationally efficient method is based on a novel technique that utilises hash tables to combine efficient density estimation, and implements sampling in a single pass of the dataset. However, this method is somewhat difficult to implement (*Palmer and Faloutsos*, 2000).

Cluster sampling

In cluster sampling, the data are allocated into groups and the sampling is then based on the random selection of whole groups, rather than taking samples from all groups. The distinction here is that the clustering does not identify homogeneous groups, but rather individual groups of data that are distributed identically to the entire database. A typical example is spatially defined groups, such as individual populations in geographical areas. An example of how cluster sampling might be grouped is in sampling hydrographs according to the occurrence of rainfall events. Time-series data corresponding to similar rainfall events may be considered equivalent groups, and therefore it may be more efficient to sample several characteristic events, rather than the complete hydrograph.

Provided that suitable groups can be defined, cluster sampling can quickly draw a representative sample, since a few groups can be equivalent to the entire database. However, the omission of entire groups can potentially reduce the representativeness of the sample taken. Cluster sampling may result in holes within the training data due to the omission of large regions of data, and consequently the trained ANN will be required to extrapolate into these regions, and is likely to perform poorly (*Sarle*, 1997).

Stratified Sampling

In stratified sampling, the database T is divided into H groups, which are referred to as *strata*, where the data within each stratum are relatively homogeneous, and distinct from data in other strata (*Cochran*, 1977; *Mulvey*, 1983; *Kpedekpo*, 1973). The formation of strata is such that all data belong to one, and only one, stratum (i.e the groups are disjoint). Hence, the stratification satisfies the condition

$$N = N_1 + N_2 + \cdots + N_H, \tag{3.3}$$

where N_j denotes the number of data within stratum j, and is commonly referred to as the stratum size (*Cochran*, 1977).

The benefits of stratified sampling over SRS are that estimates based on the sample can be weighted according to the number of data within each stratum, which can improve the accuracy of results. The method is also said to be more efficient, since for a sample of n data, estimates from a stratified sample will have a lower error than an equivalent size sample generated by SRS, which means that potentially fewer samples need to be taken. According to *Kpedekpo* (1973), effective implementation of stratified sampling considers the following:

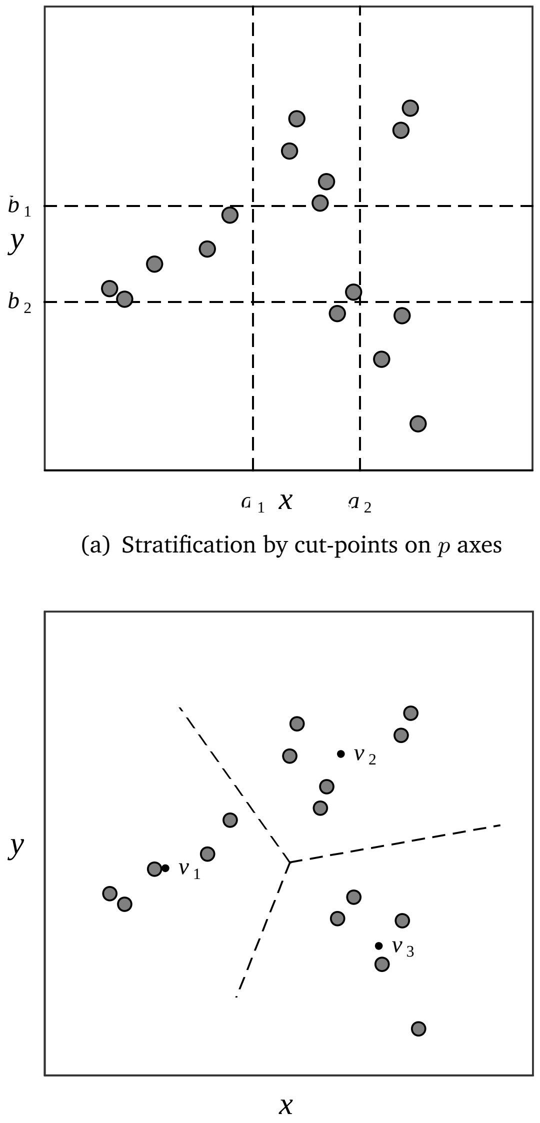
- number of strata,
- location of strata boundaries,
- allocation of samples from strata, and
- choice of stratification variables.

Figure 3.4 describes the two approaches for defining strata boundaries. The simplest way is to partition the sample by cutting each individual axis of one or more variables, as in Figure 3.4(a). In the case where variables take discrete attributes, the choice of where to locate the cut-points may be obvious (e.g. male or female sex, discrete pipe diameters etc.). However, cut-point stratification of continuous variables is more difficult. Although several methods for defining optimal strata have been devised based on the theoretical properties of sample estimates of conventional statistics (*Kpedekpo*, 1973), such as the mean and variance, similar analysis cannot be applied to non-parametric regression. Furthermore, since the cut-point stratification forms hypercubes, many empty strata may be formed and the average stratum size may decrease due to the increasing dimensionality, which could adversely affect sampling.

In the case of multivariate stratification, clustering algorithms may be applied to generate strata, which is referred to as cluster-based stratified sampling (CBSS) (*Gill et al.*, 2004). Partitioning algorithms are often used to perform the clustering, in which data are grouped according to their nearest prototype vector, as shown in Figure 3.4(b). In clustering terms, the volume surrounding a prototype vector is referred to as the *Voronoi space*, and this defines boundaries of the stratum. The CBSS approach is considered highly suited to the sampling of multivariate data for ANN development, since the partitioning of the database into homogeneous groups increases the representativeness of the sample. Stratified random sampling for ANN data selection has been described in several examples, based on partition clustering algorithms. Examples of algorithms include the k-means, self-organizing map (SOM) and the fuzzy c-means. *Svozil et al.* (1995) and *Bowden et al.* (2002) applied a partitioning of data based on the self-organizing map prior to sampling. The methodology has since been adopted in several similar ANN applications to water resources modelling (*Anctil and Lauzon*, 2004; *Zhang et al.*, 2004a; *Kingston*, 2006). *Shahin et al.* (2004) utilise fuzzy c-means clustering to partition the data, although the benefit of a fuzzy approach is only marginal, since ultimately hard (as opposed to soft) clustering is required due to the unique stratum membership constraint. Alternatively, hierarchical or agglomerative clustering could also be used to form the strata.

Regardless of the type of clustering used, the important benefit is that an optimal stratification can be obtained, which avoids the need to specify cluster boundaries (*Mulvey*, 1983) However, the challenge is that clustering algorithms typically require that the number of clusters (i.e. the number of strata) is known, or at least specified, and it is often necessary to determine a suitable number. As previously mentioned, the task of choosing the number of strata may be a simple task in the case of discrete variables, but is more difficult for continuous variables. In clustering algorithms, the same challenge also exists, and suitable methods for determining the optimal number of partitions need to be identified in order to successfully apply the CBSS approach.

The allocation of samples has also yet to be examined in detail for CBSS methods applied to ANN training data sampling. Both *Svozil et al.* (1995) and *Bowden et al.* (2002) draw a single sample for training, test and validation. Of course, this implies that there are a sufficient number of small, homogeneous groups. Consequently, the sample size and representativeness is directly a function of the number of partitions. *Svozil et al.* (1995) ensures that at least n partitions are



(a) Stratification by cut-points on p axes

(b) Stratification by clustering in p dimensions

Figure 3.4: Multivariate stratification can be achieved by (a) cut-points that divide stratification variables along each axis to form strata, or by (b) clustering the data to define strata according to naturally occurring groups.

used to draw approximately the required sample size in this manner. Conversely, *Bowden et al.* (2002) concludes that if the clustering is optimised and the data are sufficiently homogeneous, then the number of partitions essentially determines the sample size, which can result in a smaller sample. However, *Bowden et al.* (2002) concedes that determining the optimal number of partitions poses a considerable challenge. *Kingston* (2006) randomly samples all data within each partition in proportion to the desired sample sizes for training, test and validation, although little justification for this is provided. Neither of these examples have fully considered the importance of sampling from within the partitions, and the suitability of the approaches has yet to be thoroughly assessed, or compared.

The choice of stratification variables is also important, since improvements will only be observed if there are distinct groupings in the data, or groups can be found that are sufficiently homogeneous. Stratification on variables that are uniformly distributed will not yield an improvement in the quality of the sample in comparison with SRS (*Cochran*, 1977). Clustering in multiple dimensions may result in an increased number of sparse strata, which may affect sampling performance. Consequently, the choice of stratification variables may require a trade-off to optimise the benefits of stratification, by minimising stratification on marginally structured variables. Induction based stratified sampling (IBSS) is a variation of CBSS in which only the most salient variable is stratified (*Gill et al.*, 2004). It is argued that IBSS can provide some improvement by reducing the dimensionality of the clustering, although the saliency of the input variables must be known. The application of IVS filter algorithms can provide this kind of information. However, *Gill et al.* (2004) do not discuss the case where one or more variables are equally informative, which is quite possible. Given that IVS has removed both noise and uninformative variables, it is expected that unless one variable is particularly dominant, stratification on all input variables will be required to capture a representative sample of input-output data.

Multi-stage sampling

Multistage sampling generally refers to grouped sampling methods that utilise non-SRS sampling within groups, but potentially could involve any combination of single-stage methods. Common examples are stratified and cluster sampling, in which the per stratum sampling is not SRS, but some other form of sampling (often systematic) instead. The idea is that the sampling method imposed can account for heterogeneity within each cluster to yield an even more representative sample. The efficiency of this method will depend on the stratification or clustering, since homogeneous clusters will yield little additional improvement.

A multi-stage design was applied to the sampling of geotechnical data for ANN prediction of foundation settlement (*Shahin et al.*, 2004). In this case, the fuzzy c-means clustering algorithm was used to perform the stratification and the degree of cluster membership was then used to determine which samples were drawn from each stratum. Systematic samples could be drawn by taking samples in order of increasing distance from the prototype vector (i.e. from the centre of the cluster outwards) (*Bowden et al.*, 2002). However, there is no guarantee that the sample would be better than a random sample, since the distance is not necessarily related to the spatial distribution of data within the cluster. For example, two points may be the same distance from the prototype vector, but far from each other in opposite regions of the cluster. If stratification or clustering is not performed on all dimensions, multi-stage sampling can potentially offer some improvement of stratified random sampling, since there may still be some intra-cluster heterogeneity. However, provided all available dimensions are considered during the stratification stage, then the groups are likely to be homogeneous and stratified random sampling would be expected to draw an equally representative sample.

3.4.2 Non-probability Sampling

Non-probability sampling, by definition, implies that probability of selection of data is undefinable. Unlike probabilistic methods, the selection frequency of some data can be zero, and the exclusion of these data from the sample, or the impossibility of selecting certain samples by the sampling method, potentially results in bias. Non-probability sampling may be easier to implement in many cases. However, it is considered less rigorous and it is more difficult to determine the quality of the sample obtained.

Convenience Sampling

Convenience sampling refers to methods where samples are selected based on ease of sampling. In survey design or data collection, this is a common choice due to the expense or difficulty in obtaining data. However, the method is strongly biased by the factors that determine the convenience of sampling and it is not always possible to obtain representative data.

In ANN development, convenience sampling is often observed in time-series modelling. Many practitioners will use successive intervals of data for training, testing and validation, respectively. The motivation is that the time-series order of data is preserved, which makes it convenient to plot, visualise and interpret the data and model predictions. However, the danger is that trends or uncharacteristic behaviours might not be observed throughout the data record, and consequently the data samples will not be representative of each other. For example, a hydrological database might contain data for a recent flood event that is used as test data, where older data were used for training. Consequently, the model will perform poorly due to a lack of sufficient training data.

Judgement Sampling

In the case where some expert knowledge is available regarding the system or process under consideration, it is often considered appropriate to use this additional information to guide the selection of training data. In environmental time-series modelling applications, it is not uncommon for judgement sampling to consider seasonality of the data, where data spanning one or more whole seasonal cycle are used as training data, and data from other corresponding seasonal cycles are used as test and validation data. However, without any data analysis to support the sampling, there is the potential for errors in judgement to create bias. It may be the case that other unknown patterns exist in the test or validation data that are not contained in the training data due to other states of the system.

Quota Sampling

Quota sampling is often used in the case where data may belong to one of a set of c classes, and it is required that specific instances or proportions of each class are included in the sample. Quota sampling overlaps several other sampling techniques, and can be considered a special case of multistage sampling involving stratification and judgement sampling. Stratification groups the data according to class, and then a specified quota from each class are sampled based on expert judgement.

Quota sampling is often used for survey design and a common example is the 50:50 sampling of gender in surveys. The drawback is that the sample is likely to be highly biased by the defined quotas for each class. The method also implies that there is some extensive knowledge of the classes within the data, and a rationale behind the formulation of quotas. The bias and non-randomness are considered to be a disadvantage in comparison to stratified random sampling, which can essentially achieve a similar sample with less bias, and less *a priori* knowledge of the data.

Systematic Sampling

Systematic sampling is a non-probability sampling method, in which a starting point is selected and then every k^{th} sample onward is selected (*Cochran*, 1977). Given a sample size, n, the sampling interval $k = N/n$ is determined. The first sampling location is chosen by drawing a random location $m \in [1, k]$ and then sampling locations $m + k, m + 2k, \ldots$ etc. If the data are unordered, then systematic sampling effectively yields a uniformly random sample. Systematic sampling is therefore an even more efficient means of implementing SRS, since there is no need to generate random numbers.

Systematic sampling can also be used to efficiently implement a form of stratified sampling. If the data are somehow sorted, then systematic sampling is implicitly stratified due to the alignment between the sampling interval and the structure within the data. *Baxter et al.* (2000) used systematic stratified sampling for sampling ANN training data for a water quality model, in which the data were sorted in order of the output variable.

The drawback of systematic sampling is that if the data are unknowingly ordered, or if care is not taken during sampling, then sampling can be biased due to structure in the data. An example is the systematic sampling of a periodic time-series, in which the sampling interval coincides with the period of the data. In this case, the sample will contain only data corresponding to the same point within the period.

Duplex Sampling

Kennard and Stone (1969) developed the CADEX and DUPLEX data splitting algorithms for split-sample validation of regression, and these approaches are sometimes collectively referred to as Kennard-Stone (K-S) data splitting. CADEX initially selects data that lie farthest from all other points within the database. Training and calibration data are alternately selected by selecting data that lie farthest from any previously selected points. DUPLEX is a modified form of CADEX in which data are selected in pair-wise manner in order to reduce the optimism of the test data (*Snee*, 1977), which proceeds as follows:

DUPLEX:

i. Find x_i and x_j that maximise $\|x_i - x_j|$ and move from T to training set.

ii. Find x_i and x_j that maximise $\|x_i - x_j|$ and move from T to test set.

iii. Find next sampled pair x_i and x_j, such that $|x - c|$ is maximised, where c is the centroid of the sample S

iv. Repeat, alternating allocation to training and test samples, until smallest set is filled.

Here, in order to initialise each sample, the initialisation sequentially finds the pair of data that lies farthest from each other within the database; the first pair allocated to training, the second to the test data.

The approach has zero variance, as it is fully deterministic and only one split is possible for any given database. Although Kennard-Stone sampling generates a 50:50 split into two datasets, it is possible to generate data sets of arbitrary proportions by allocating data to the smaller set until it is filled, and allocating all remaining data to the larger set (*Snee*, 1977). Both CADEX and DUPLEX algorithms have been used widely in the field of chemometrics, including several applications to ANN development (*Despagne and Massart*, 1998). However, in comparison to other approaches, they are relatively unknown within the field of water resources or environmental modelling and analysis.

Supervised Sampling

Supervised sampling techniques describe sampling methods where data are selected based on some criteria regarding the quality of the sample. The allocation of data into samples is treated as an optimisation problem, in which the aim is to ensure that the samples are maximally representative (or, minimally dissimilar) of each other so that the bias in model validation due to the hold-out is minimised. The two most important considerations for this approach are the method for iteratively searching through the many combinations of samples, and the objective function that evaluates the similarity of the samples.

Bowden et al. (2002) notes several examples where supervised sampling has been achieved using a manual trial-and-error approach to optimally allocate samples. However, very few details are given as to the exact implementation of the methods used, and consequently, these are difficult to reapply. Furthermore, although statistically similar datasets may be found that satisfy hypothesis tests, the result may not be optimal given that only relatively few combinations will be explored. Consider sampling of n data into three disjoint samples of sizes a, b and c. The number of unique combinations of samples is

$$C_a^n C_b^{n-a} C_c^{n-a-b}, \tag{3.4}$$

where $a+b+c \leq n$. Exhaustive search is not feasible for large datasets, since the number of possible samples that could be generated is very large. Random search could be applied to automate the procedure. However, given the size of the search space, a large number of trial samples will need to be evaluated, otherwise the result may not be optimal given that only relatively few combinations will be explored.

A natural refinement of the random search approach is the application of heuristic optimisation algorithms, such as genetic algorithms. The potential application of genetic algorithms (GA) to this sampling problem has been demonstrated by both *Reeves and Taylor* (1998) and *Bowden et al.* (2002). The former use the recombination and reassortment (RAR) operator (*Radcliffe*, 1993) as the basis for cross-over and mutation, which is designed for GAs applied to number set problems. Given two samples of length n, RAR places ω copies of data that occur in both samples, and one copy of data that are unique to either sample, into a pool or *bag*. A new sample is then drawn by simple random sampling of n data from the pool, ensuring that the sample does not contain multiple copies. Using the RAR operator, the relative degree of recombination can be set to tune the convergence characteristics of the algorithm. The parameter w (usually set to 1 or 2) controls the relative degree of recombination to reassortment. A high value of w increases the probability of sampling data that are common to both original samples and leads to faster convergence. So far, the RAR GA has only been considered for the selection of data for training and out-of-sample testing. Although it seems highly suitable, the application to the selection of three samples for training, testing and validation has yet to be examined.

Bowden et al. (2002) utilised a more conventional floating-point GA with a random number seed, which generates a unique pseudo-random sort-key to permute the ordering of data, as the single decision variable. Samples are generated by fixed, contiguous partitions of the required size, where the composition of the samples depends on the order of data. However, the formulation seems inefficient in comparison with the RAR GA for several reasons. First, each random number seed gives a unique sort key, but does not strictly result in a unique permutation of the data. Second, unique permutations of the data themselves do not necessarily give rise to unique samples, since the ordering may simply reorder data within the same partitions. Consequently, many unique number seeds will correspond to equivalent samples, reducing the efficiency of the search. Finally, and perhaps more importantly, is that the encoding of the optimisation problem in this manner does not utilise the full potential of GAs, since there is, in fact, no functional relationship between the random number seed value the composition of the sample. The fundamental basis for evolutionary search strategies is that characteristics of good solutions are preserved through subsequent iterations to allow convergence as a result of cross-over and selection. Consequently, the algorithm developed by *Bowden et al.* (2002) only finds an optimal solution due to the random search behaviour that results from cross-over and mutation.

Formulation of the objective function is a critical consideration in the formulation of a GA, as it mathematically defines the notion of quality. Various methods for evaluating the sample quality have been proposed in order to define a suitable objective function. *Reeves and Taylor* (1998) evaluated the quality of the training sample based on the validation performance of a trained model. In this sense, the method for evaluation is similar to wrapper algorithms for input variable selection (see Section 2.3.4). Here, the drawbacks of computational efficiency and the need to develop many ANN models are the same.

Statistical analysis can be applied to define the quality of the sample, for which similar comparisons can be made to IVS filter algorithms. The definition of quality is invariably based on the representativeness of training, test and validation data, which is usually defined by measuring the statistical similarity of the samples (*Bowden et al.*, 2002). Conventional statistical tests include the t-test for the similarity between the means of samples, and the analysis of variance (ANOVA) F-test for comparing the spread of data samples. However, both the t and F statistics are highly influenced by the centres of the respective distributions, and as such do no provide detailed information regarding the tails of the distribution.

The Kolmogorov-Smirnov (K-S) statistic has also been used to compare the distribution of the samples (*Bowden et al.*, 2005). The K-S statistic measures the similarity as the maximum deviation between the cumulative frequencies of two distributions. The K-S statistic is considered a robust method of comparing two samples, since it considers the entire distribution. A similar criticism to the conventional t and ANOVA F tests can be made of the Kolmogorov-Smirnov (K-S) statistic, which is known to favour matching of the mean of two distributions where the rate of change in the cdf is the greatest. Given that the sampling variance is often due to poor sampling in the sparser regions of the data, it would seem that the application of statistics based on the mean could potentially provide a false indication of the representativeness of the data. The Andersen-Darling statistic (*Stephens*, 1974) might prove a better choice, as it places more emphasis on matching the tails than the K-S test.

Regardless of the statistical test used, there are several hitherto unresolved issues with the GA sampling approach. First, each dimension of the data is invariably considered separately, and therefore only univariate statistics are computed. It is assumed that statistical similarity, as inferred by univariate hypothesis tests on each dimension, corresponds to the selection of truly representative samples. The sampling method should consider obtaining representative samples of the joint distribution, that is to say, there is equal representation of all input-output tuplets in the training, test and validation samples. Since a suitable investigative study has yet to be undertaken, there has been no conclusive evidence provided that optimisation of statistical similarity measures produces optimal data samples. It has been demonstrated that the objective function produces similar training, test and validation samples, but there is no evaluation of their quality in terms of bias, or variance. Furthermore, the deliberate sampling of validation data (by including properties of the validation data in the objective function) suggests that these studies are optimistically biased, and an out-of-sample test is necessary to truly validate the approach.

3.5 Comparison of Approaches

Table 3.1 presents a qualitative comparison of the different approaches to sampling ANN data. Several criteria are used to differentiate between the different approaches based on the quality of the sample, and the effort required to draw a sample. In terms of quality, the bias and variance characteristics of samples are considered to reflect the ability of the sample to consistently draw good samples. Differences between training, test and validation samples will also be manifest as bias (i.e. model error). Representativeness refers to the ability of the sampling method to provide coverage of all parts of the modelling domain. Computational constraints may also be important, and in Table 3.1, the two criteria considered are speed and scalability. Methods differ in speed, but there will also be variation in the increase in computational effort as the number of samples to be drawn increases, or the length of the available dataset increases. Finally, the amount of *a priori* information regarding the data is important, as this will determine how easily different methods can be implemented. Some methods can be implemented without any knowledge, such as SRS; but other methods, like importance sampling, may only be implemented based on pre-existing or determined knowledge of the available data.

Simple random sampling (SRS) is the most basic, and possibly the easiest prob-

Table 3.1: Qualitative comparison of sampling methods

Method	Sample Quality			Computation/Effort		
	Bias	Variability	Representativenes	Knowledge	Speed	Scalability
SRS	Low	High	Poor	None	Fast	Linear
DBS	Low	Low	Good	High	Slow	Exponential
Importance	Med	Low	Good	High	Fast	Linear
Stratified Random	Low	Low	Good	Low	Fair	Linear
Systematic (Random)	Med	High	Poor	None	Fast	Linear
Systematic (Stratified)	Med	Low	Fair	Low	Fair	Squared
Judgement	High	None	Poor	High	Fast	N/A
Convenience	High	None	Poor	Low	Fast	N/A
Trial-and-error	High	Low	Fair	None	Slow	Exponential
Kennard-Stone	High	None	Good	None	Slow	Exponential
GA sampling	High	Low	Fair	None	Slow	Exponential

ability sampling technique. On average, SRS will not yield a good sample, due to the poor representation of non-uniformly distributed data. The method is unbiased in the traditional sense, that any sample has an equal chance of being selected. However, the quality of the sample has a high variance, and can be very poor. Despite the widespread application of SRS to ANN data sampling, it is not a good choice, as it is unable to reliably generate a good hold-out sample for any real-world application, for which data are naturally distributed (*Palmer and Faloutsos*, 2000).

Convenience sampling and judgement have often been used to select training data for ANN development. The methods are very quick to implement, since there is essentially little to no computation required. However, there is no easy way to determine the quality of the sampling, since the amount of bias due to sampling is generally unknown, although with both methods, the bias is usually significant. In this sense, the bias is due to the selection of specific data, and exclusion of others, which leads to a difference between the ANN model based on the sample and the true model. Transferral of the method from one application to another is also very difficult, as it relies on expert understanding of the idiosyncrasies of the problem at hand, making the use of judgement a necessity.

Genetic algorithms or heuristic search can guarantee that a near-optimal sample can be drawn, to ensure representativeness. However, the sampling is heavily biased by the objective function that is used to evaluate sample quality, and it is difficult to estimate the impact of such bias. Wrapper-like approaches to sampling ANN data are computationally intensive and are undesirable for the same reasons wrappers are less favoured than filters for IVS (see Section 2.4). However, methods for the statistical assessment of sample quality are possibly insufficient to determine true sample quality, nor has a conclusive demonstration of sample similarity and sample quality been established.

DBS is one of the better sampling methods for ANN development. DBS can be used to increase the frequency of sampling for sparse data so that the variability of sampling is reduced, which has been demonstrated to improve the performance of ANN models for the prediction of rare events. However, the drawback of the approach is that significant computational effort may be required to estimate the necessary pdf for determining the sample probabilities. Although some computational improvements can be implemented for discrete data, for continuous data the DBS method will only be a feasible option if the pdf of the data is already known, or the dimensionality and length of the data is sufficiently low to permit density estimation.

Stratified sampling is one of the best methods for drawing a quality sample. The stratification partitions the data so that samples are drawn from the entire data space, which ensures that the data are representative and the ANN will less likely need to extrapolate during testing or validation. Cluster-based stratified sampling has been found to be ideally suited for the development of ANN models. The scalability of the algorithm relates more to the clustering than the sampling, and provided the number of clusters is kept small, the computational effort required to perform stratification will be feasible. The computational requirement of clustering is a significant factor in the overall effort, but is typically less intensive than the pdf estimation required for DBS. Despite the several demonstrations of CBSS of ANN training data, the issues of determining the number of strata and how to best draw samples from within them are hitherto unresolved.

Chapter 4

Synopsis of Publications

> *"Hackworth was a forger, Dr. X was a honer. The distinction was at least as old as the digital computer. Forgers created a new technology and then forged on to the next project, having explored only the outlines of its potential. Honers got less respect because they appeared to sit still technologically, playing around with systems that were no longer start, hacking them for all they were worth, getting them to do things the forgers had never envisioned."*
>
> Neal Stephenson, *The Diamond Age* (1995)

This chapter discusses the contributions of the six publications presented in subsequent chapters, which form the core of this thesis. Overall, this thesis is focussed on increasing understanding of the methodology for ANN development and Figure 4.1 illustrates the relevance of the publications and their context within a framework for ANN model development. In particular, two aspects of ANN development are examined: input variable selection (IVS), and data splitting. As highlighted in the review of literature (Chapters 2 and 3), the importance of both of these issues for ANN model development is arguably greater than for conventional modelling approaches, and can severely impact on model performance. Despite the emergence of rules and considerations to guide other aspects of ANN development, it is apparent that there are significant, yet unresolved issues in these areas, which are consequently addressed in the publications presented herein.

The important issue of IVS is discussed in Publications 1, 2 and 3; which follow on from the review of IVS approaches in Chapter 2. Central to these papers is the IVS approach of *Sharma* (2000) described within the literature review, and the issues discussed with regard to the reliability and accuracy of the selection, and

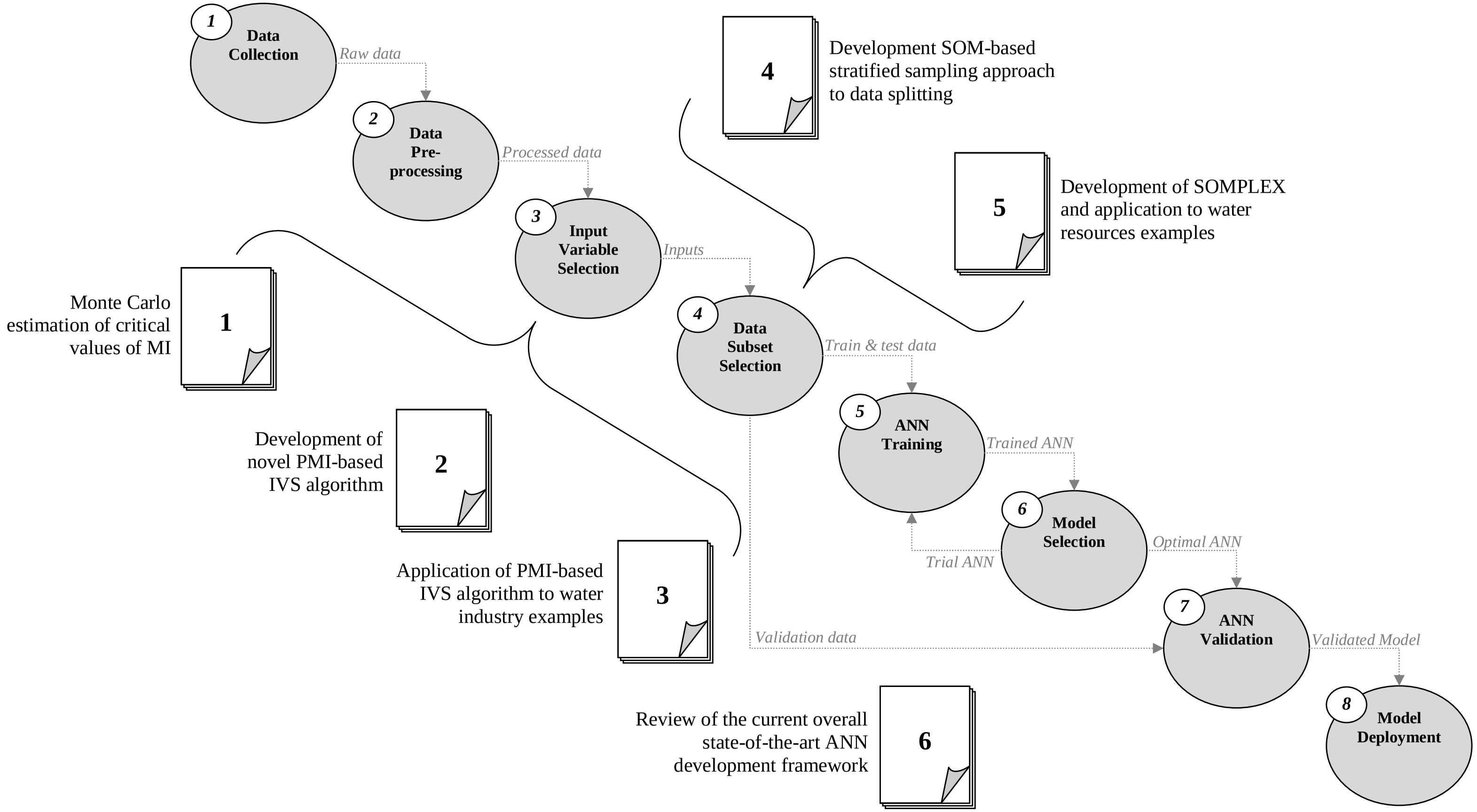

1
Data Collection
Raw data
2
Data Pre-processing
Processed data
3
Input Variable Selection
Inputs
4
Data Subset Selection
Train & test data
5
ANN Training
Trained ANN
6
Model Selection
Trial ANN
Optimal ANN
7
ANN Validation
Validation data
Validated Model
8
Model Deployment
4
Development SOM-based stratified sampling approach to data splitting
5
Development of SOMPLEX and application to water resources examples
1
Monte Carlo estimation of critical values of MI
2
Development of novel PMI-based IVS algorithm
3
Application of PMI-based IVS algorithm to water industry examples
6
Review of the current overall state-of-the-art ANN development framework

Figure 4.1: Contribution of publications presented in this thesis within the context of ANN development

the computational requirements. Publications 1 and 2 are concerned with the theoretical development of an improved implementation of the algorithm, and Publication 3 provides a contextual application of the algorithm.

Publication 1 describes a Monte Carlo simulation (MCS) approach for determining critical values of the kernel density estimation of MI that underpins the PMI-based IVS algorithm. This paper presents the motivation for adopting MCS as a pragmatic solution in the absence of a theoretical expression for the error in finite-sample estimates of MI, and illustrates the use of MCS-based confidence bounds in the IVS algorithm. In this case, termination is based on the evaluation of the significance of MI measured between a candidate input and the output. Sharma's original implementation relied upon a bootstrap estimation of MI estimation error, and consequently the algorithm was shown to potentially be both computationally intensive and unreliable. The IVS example given in the paper clearly demonstrates the behaviour of the more robust MCS-based estimates of confidence bounds, which are more stable; in comparison to those based on a relatively small and computationally intensive bootstrap, which are highly variable and therefore unreliable.

Estimation of critical values of MI is a central issue in applications that utilise MI as a measure of relevance. A major contribution of this work is the production of a table of critical values of MI computed for the case $I(x;y)$. The computation of MCS estimates necessary to generate these tables represents a considerable usage of CPU time, especially for estimates where the sample length becomes large, and so these tables provide a reference for future work. Previously unpublished tables of MI estimates for multivariate cases were also computed during this research, and are presented in Appendix A of this thesis. An important result is the quantification of the increase in errors for MI estimates of increasing dimensions. This result adds further justification for IVS based on bivariate PMI estimates, rather than multivariate joint mutual information (JMI), because multivariate estimates of JMI have significantly greater uncertainty; which would potentially obscure relationships between variables.

Publication 2 presents the development of the improved version of the IVS algorithm, which is based upon the estimation of partial mutual information (PMI). The paper describes the development and evaluation of the algorithm using several alternative criteria for terminating the selection process. Three alternative termination criteria are formulated:

1. termination using off-line estimates of MI confidence bounds determined using Monte Carlo simulation;

2. termination using the Akaike information criterion, which finds the optimum trade-off between dimensionality and information content in the set of input variables; and
3. use of the Hampel outlier test to terminate selection when the remaining candidates are equally insignificant, and presumably irrelevant.

The first criterion, developed in Publication 1, is potentially limited by an assumption of Gaussianity in the data used to estimate the critical values of MI. Consequently, the second two criteria provide alternative approaches to terminating selection, which each address this potential limitation.

Publication 2 also describes the evaluation of algorithm performance using each termination criterion, based on the selection of inputs for a suite of linear and non-linear datasets. The paper critically evaluates Sharma's original implementation, and quantifies the sensitivity of the algorithm to the bootstrap size used. Resulting comparisons with the original implementation show clearly that the alternative criteria can significantly improve the ability of the algorithm to select the optimal set of input variables. Furthermore, by avoiding the bootstrap, the computational effort involved in performing the selections is reduced by the order of 90%. Overall, each of the criteria presents a solution to implementing the algorithm, without requiring a trade-off between accuracy and effort.

Importantly, the use of linear relevance measures, such as correlation, in IVS filter design is conclusively shown to be a flawed approach when developing ANN models through comparisons of the performance of a correlation-based approach and the MI-based approach. The correlation-based approach only identifies the correct input variables when linear relationships are present, and fails to identify all variables when the dependence is non-linear. This is considered an important result to emphasise, since the use of MI comes at considerable computational expense, in comparison to correlation. However, although more computationally intensive, the use of MI is obviously more appropriate than linear correlation when the underlying assumption of non-linearity is implicit to the development of ANN models.

In Publication 3, the application of the PMI algorithm to ANN model development is demonstrated within the context of two water quality modelling examples: an ANN meta-model of a well-known water distribution system simulation, and an ANN model to forecast water quality within a real-world water distribution system. Importantly, this paper provides a comparison between the approach to model development using the PMI-based algorithm for IVS, and an existing state-of-the-art approach (Chlorcast©), which adopts a greedy approach by selecting all inputs within a given time window. In each of the examples, the IVS approach is shown to be able to identify an optimal subset of input variables for the ANN, and therefore produce a more parsimonious ANN with an equivalent prediction performance. Furthermore, this paper demonstrates how the analysis of selected input variables can provide an understanding of the relative importance of relationships that reside within the data, and how the ANN makes predictions; which is otherwise obscured by the "black-box" approach.

Publications 4 and 5 examine the problem of data splitting, which is the fourth stage of the ANN model development framework, as shown in Figure 4.1. As stated in the literature review in Chapter 3, hold-out validation is the most common methodology used in ANN development to avoid over-training the network and ensure good generalisation. Methods for generating training, test and validating data vary within the literature, and as with IVS, there is little consensus of what is an appropriate technique for a given ANN application. In many cases, the importance of data splitting is understated or overlooked completely. In other cases, although the data splitting approach is described, there is no justification for the approach given, and no assessment of the impact on model performance. Consequently, there is a need to quantify and compare the utility of various sampling algorithms used for ANN development.

Publication 4 presents an investigation of a data splitting method based on stratified random sampling of the self-organizing map (SOM). Several examples within the literature have suggested the SOM as a tool for selecting training, test and validation data. Several of the reported applications within the literature highlight the potential benefits of this approach, which is essentially a form of stratified sampling. The experimental study presented in Publication 4 considers the specific implementation of so-called SOM-based stratified sampling (SBSS), given that the applications to date utilise different variations on the approach, which differ with respect to how data are drawn from the SOM. Additionally, the paper considers the impact of SOM attributes, namely the size of the map, on the quality of the data sets that are produced; and also the contribution of random initialisation of the SOM to variance in the samples.

Comparisons between different SOM-based methods are made in Publication 4 to show that some of the approaches within the literature tend to produce subsets that generate models that exhibit high variance and bias in the test and validation error. These include drawing a single random sample, and drawing data in proportion to the cluster size. On the other hand, the Neyman allocation rule, which had previously been untested, is found to significantly reduce both the bias and variance of the SBSS approach. The contribution of SOM initialisation is found to be negligible, and could be eliminated as a significant source of the observed variation in sample quality.

The experiments undertaken in this research demonstrate that the application of cluster validity indices (CVIs) to determine the optimal SOM size is potentially flawed. Experiments indicate that larger grid sizes tend to result in good sampling, whereas CVIs tend to suggest one or two clusters. It is evident in the research undertaken that an optimal grid size exists, which minimises the variability and size of model error, which is much larger than that predicted using CVIs. The problem with using CVIs is that datasets in regression are not always strongly clustered globally, nor is global clustering relevant to sample quality. Moreover, the partitioning required to ensure representation of the data with reasonable precision is more dependent on local structures. The size of SOM specified according to a popular heuristic rule was found to correspond closely to the optimal grid size, in terms of sample quality, which offers a more convenient way to determine the grid size to implement SBSS.

Finally, and arguably more importantly, the quality of the sample drawn using the SBSS approach was benchmarked against several other popular approaches from the literature. To the author's knowledge, a comparison of multiple sampling strategies such as this has not been undertaken prior to this research. This is an important comparison to make, since many reports of ANN development include unsupported claims of superior data splitting based on the reported technique. These results also support the qualitative comparisons that are made within Chapter 3. Furthermore, this study has introduced the DUPLEX algorithm, which is a popular deterministic data splitting algorithm in analytical chemistry, but is rarely used in ANN water resources modelling applications. Although this algorithm is traditionally used to generate only train and test sets, this paper illustrates how DUPLEX can be used to generate the three datasets required for training, testing and validating ANN models.

On the basis of the comparative study in Publication 4, some broad guidelines were developed for choosing a sampling technique for data splitting. It is evident that the distribution of data will influence the relative effectiveness of different sampling techniques. First, the case against simple random sampling (SRS) is clear and as a general rule should not be used for data splitting—especially when the distribution of available data is skewed in some way (which is generally the case in real-world datasets used in model development). SBSS provides a sampling technique that generates a more balanced data splitting approach for multivariate data, as it takes into account the distribution of data. However, due to the inherent issues related to the specification of the SOM size and learning parameters, the SBSS approach can be difficult to implement. Two simpler approaches might be considered more suitable in the case of lower dimensions, or when data are less skewed. In particular, Systematic stratified sampling, that is, systematically sampling data that are sorted along the output variable, was found to yield a simple and effective way of implicitly stratifying the sample, but is only justified for datasets of one or two dimensions. The DUPLEX algorithm (*Snee*, 1977), which selects data based on maximal distance, was found to outperform SBSS. The approach is fully deterministic (i.e. no variation) and is relatively straightforward to implement and can be applied to any dataset. However, the DUPLEX algorithm is restricted to moderately sized datasets, due to its poor computational scalability.

Publication 5 presents the development of a novel hybrid SOM-DUPLEX data splitting algorithm, which is called SOMPLEX. The combination of SOM clustering with DUPLEX intra-cluster sampling is found to provide an excellent approach to data splitting. The DUPLEX sampling can overcome the variation in performance due to random sampling of large clusters when using the SBSS approach, and improves the overall reliability of the data splitting approach. The conventional DUPLEX algorithm also has high computational complexity, and the computation requirement grows rapidly with increasing number of data. Consequently, the additional benefit of the SOM clustering is to significantly reduce the computational requirements.

Publication 5 also validates the arguments presented previously in Publication 4 by considering the application of data splitting algorithms to real-world applications in water resources, including: salinity forecasting, chlorine forecasting and coagulation process modelling. This presents the first example of the use of the DUPLEX algorithm within a water resources context, where the algorithm generated good results. Comparisons of bias and variance for resampled training, test and validation data are used to further reinforce the benefits of the SOMPLEX approach.

Publication 6, the final publication in this thesis, presents an overall review of ANN model development. Despite a history of applications dating back to the mid-1990s, the application and development of ANN models is still relatively new to the water industry. The vast amount of ANN literature can intimidate or confound many practitioners. The contribution of this book chapter is to provide some guidance in the area of ANN development, and disseminate the knowledge and insights gained through this research.

The overall motivation for this research was to improve techniques for ANN development, and to provide guidance on appropriate modelling choices. A framework is proposed, summarising the current state-of-the-art in ANN development. The framework is presented as a data-flow diagram, and better illustrates the flow of data through the stages of model development, including the iterative feedback loops during model training. The ANN framework also highlights the interaction and choices available to the modeller at each stage of development. Although this thesis focussed on input variable selection and data splitting, it is important to emphasize that ANN development is a series of stages, from data collection through to model validation, and that each of these steps can influence the quality of model development.

Much of ANN literature is concerned with comparisons between ANN architectures, and the development of novel learning paradigms for ANN training. The main viewpoint presented in Publication 6 is that although machine learning paradigms and ANN architectures are undoubtedly important areas of research, the specific *flavour* of ANN or training algorithm are not necessarily the foremost consideration during ANN model development. Indeed, many ANN architectures are fundamentally similar and will most likely give satisfactory results, provided that the auxiliary stages of development are implemented correctly. It is how the ANN model is developed—namely the consideration of data collection, data preprocessing, input variable selection, data splitting and performance criteria—are likely to have a more significant impact on the utility and quality of the ANN model that is developed, consequently it is these stages of development that should be the foremost concern.

Chapter 5

Critical values of a kernel density-based mutual information estimator

Publication 1

Publication Details

This paper was presented at the International Joint Conference on Neural Networks, on July 20, 2006, as part of the special session on Information Theoretic measures. The full citation is as follows:

> May, R. J., G. C. Dandy, H. R. Maier, and T. M. K. G. Fernando, Critical values of a kernel density-based mutual information estimator, in proceedings of *International Joint Conference on Neural Networks*, Vancouver, Canada. July 16–21, 2006.

Although the manuscript has been reformatted in accordance University guidelines, and sections have been renumbered for inclusion within this thesis, the paper is otherwise presented herein as published.

Statement of Authorship

May, R. J. (Candidate)
Conceptual development, experimental design and implementation, interpretation and analysis of results, manuscript preparation and corresponding author.

Signed: .... Date: ....

Maier, H. R.
Project supervision and review of manuscript.

Signed: .... Date: ....

Dandy, G. C.
Project supervision and review of manuscript.

Signed: .... Date: ....

Fernando, T. M. K. G.
Review of manuscript.

Signed: .... Date: ....

May, R. J., Dandy, G.C., Maier, H.R. and Fernando, T.M.K.G. (2006) Critical values of a kernel density-based mutual information estimator, in *Proceedings of International Joint Conference on Neural Networks, Vancouver, Canada. July 16–21, 2006, pp. 4898 – 4903*

Chapter 6

Non-linear Variable Selection for Artificial Neural Networks Using Partial Mutual Information

Publication 2

Publication Details

This work has been published within the journal *Environmental Modelling and Software* as the following article:

> May, R. J., G. C.Dandy, H. R Maier, and T.M.K.G. Fernando, Non-linear Variable Selection for Artificial Neural Networks Using Partial Mutual Information, *Environmental Modelling and Software*, 23, 1312–1326, 2008.

Although the manuscript has been reformatted in accordance University guidelines, and sections have been renumbered for inclusion within this thesis, the paper is otherwise presented herein as published.

Statement of Authorship

May, R. J. (Candidate)
Conceptual development, experimental design implementation evaluation, analysis of results, manuscript preparation and corresponding author.

Signed: .... Date: ....

Maier, H. R.
Research supervision and review of manuscript.

Signed: .... Date: ....

Dandy, G. C.
Research supervision and review of manuscript.

Signed: .... Date: ....

Fernando, T. M. K. G.
Conceptual development and review of manuscript.

Signed: .... Date: ....

Abstract

Artificial networks (ANNs) have been widely used to model environmental processes. The ability of ANN models to accurately represent the complex, non-linear behaviour of relatively poorly understood processes makes them highly suited to this task. However, the selection of an appropriate set of input variables during ANN development is important for obtaining high-quality models. This can be a difficult task when considering that many input variable selection (IVS) techniques fail to perform adequately due to an underlying assumption of linearity, or due to redundancy within the available data.

This paper focuses on a recently proposed IVS algorithm, based on estimation of partial mutual information (PMI), which can overcome both of these issues, and is considered highly suited to the development of ANN models. In particular, this paper addresses the computational efficiency and accuracy of the algorithm via the formulation and evaluation of alternative techniques for determining the significance of PMI values estimated during selection. Furthermore, this paper presents a rigorous assessment of the PMI-based algorithm, and clearly demonstrates the superior performance of this non-linear IVS technique in comparison to linear correlation-based techniques.

6.1 Introduction

The development of statistical models is a well established technique for representing, and even predicting, the dynamic state of environmental systems. In the case of many environmental systems, there is an abundance of available data for model development, but a relatively poor understanding of the complex underlying processes that generate the observed system dynamics, and this favours the statistical modelling paradigm. In particular, the application of artificial neural network (ANN) architectures to environmental modelling has become widespread in recent years (*Maier and Dandy*, 2000; *Maier*, 2006). This has been mainly due to increased recognition of their superior ability to represent complex, non-linear behaviour in comparison to more conventional modelling techniques.

The development and use of ANNs for environmental modelling and data analysis has received much attention. Some examples include: real-time forecasting of air quality (*Finardi et al.*, 2008; *Pires et al.*, 2008; *Al-Alawi et al.*, 2008; *Ionescu and Candau*, 2007; *Dutot et al.*, 2007; *Sousa et al.*, 2007), ecological modelling and remote sensing (*Iglesias et al.*, 2007; *Shanmuganathan et al.*, 2006), modelling of methane biogas production (*Ozkaya et al.*, 2007), modelling and control of wastewater processes (*Raduly et al.*, 2007; *Machon et al.*, 2007) and wastewater networks (*Darsono and Labadie*, 2007), water treatment process control (*Maier et al.*, 2004), and water quality forecasting within rivers (*Alp and Cigizoglu*, 2007) and distribution systems (*Serodes et al.*, 2001; *Rodriguez and Serodes*, 1999).

Various frameworks have been proposed within the literature for the development of ANN models, based on their application to a range of environmental systems (*Maier and Dandy*, 2000; *Dawson and Wilby*, 2001; *Bowden*, 2003; *Kingston*, 2006). In particular, a common component within these emergent frameworks is the selection of an appropriate set of input variables from within the available data. However, although there is a well-justified need to carefully consider input variable selection (IVS), there is currently no consensus on how this task should be undertaken. Many of the described methods for IVS are based on trial-and-error, heuristics, expert knowledge, statistical analysis, or a combination of these. The statistical approach appears to offer an efficient methodology that is not confined to specific applications. Hence, there is a potential for suitable algorithms to become an integral component within a more robust framework for ANN development—a framework that relies more upon analysis of the data, and less upon heuristics and expert knowledge; and hence, is more in keeping with the overall ANN modelling paradigm (*Maier*, 2006).

This paper is focused on the use of a recently proposed algorithm for non-linear

IVS based on the estimation of partial mutual information (PMI). Originally proposed by *Sharma* (2000), the algorithm is highly suited to the development of ANN models due to the inherent properties of mutual information (MI); and is one of only a few non-linear IVS algorithms reported for the development of ANN models in environmental modelling applications (*Bowden et al.*, 2005). However, it should be noted that although the motivation in this paper is primarily ANN development, the nature of the proposed IVS algorithm is such that it can be used to identify inputs for any class of regression. This paper describes improvements in the existing algorithm achieved through the formulation of alternative termination criteria to improve computational efficiency and accuracy of the algorithm. It also provides a rigorous assessment of the ability of the PMI-based algorithm to outperform linear correlation-based IVS techniques when applied to non-linear systems.

The remainder of this paper is structured as follows. Section 6.2 provides a theoretical overview of IVS and the estimation of PMI, which leads to the algorithm that is subsequently described. Section 6.3 presents the formulation of several alternative termination criteria for the IVS algorithm, and Section 6.4 describes a benchmarking study, in which the termination criteria were evaluated based on the application to experimental IVS problems. Results of the study are presented in Section 6.5, and concluding remarks are given in Section 6.6.

6.2 Theoretical Overview

6.2.1 Input variable selection techniques

The IVS problem is defined as the task of appropriately selecting a subset of k variables, S, from an initial candidate set, C, which comprises the set of all potential inputs to a model (i.e. *candidates*). Defining what constitutes an appropriate subset of input variables takes into consideration the effect the choice of input variables ultimately has on the performance of models that are either incorrectly over-specified or under-specified.

An inaccurate model results when the input set is under-specified, as the selected variables do not fully describe the observed behaviour within the system under consideration. On the other hand, the inclusion of input variables that are either *irrelevant* or *redundant* (i.e. over-specification) increases the size of the model. This not only adds to the data processing time required for model development and deployment, it also adds noise, rather than information, to the model inputs and thus reduces the accuracy of the model. Furthermore, as the dimensionality of the input variable set increases for a given model, the number of data samples required for training increases exponentially. This may pose a difficulty for practitioners with limited data, or may prohibitively increase the computational requirement of model development and deployment. Given these considerations, an appropriate set of model inputs is considered to be the smallest set of input variables required to adequately describe the observed behaviour of the system.

Algorithms for IVS can be considered broadly as either *wrapper* or *filter* algorithms. Wrappers essentially treat the selection of inputs as an optimisation of the model structure. The optimisation compares and evaluates either all, or a subset of, the possible input sets and selects the set that yields optimal performance of the calibrated model. Implementation of IVS wrappers can be achieved in several ways, including: forward selection, where the input set increases from a single input until model performance is no longer improved; backward elimination, where the input set initially includes all candidates and candidates are removed one at a time; or global optimisation (e.g. evolutionary ANNs and genetic programming), where the decision to include each input is encoded as a variable within the overall model optimisation.

Considering the combinatorial nature of the IVS problem, the number of possible subsets that could be selected from a set of d potential input variables is equal to $\left(2^d - 1\right)$. The computational requirement of the trial calibration and evaluation of a potentially large number of models is considered to be a weakness of using IVS wrappers (*Kwak and Choi*, 2002; *Chow and Huang*, 2005). Furthermore, optimal performance of the trained ANN does not strictly imply optimality of the input set, since this is also dependent on additional factors such as the type of ANN architecture, training algorithm, and the performance criteria adopted. The appropriateness of a set of inputs obtained for a particular model architecture is therefore not guaranteed for another, and restricts the applicability of any input set obtained using a wrapper technique (*Battiti*, 1994).

In contrast to the *model-based* wrapper approach, *model-free* filter techniques utilise a statistical measure of the degree of dependence between the candidates and output variables as the basis for input variable selection. This separation of the IVS task from the model calibration and selection tasks not only yields a more efficient algorithm overall, but the resulting input set has wider applicability to different model architectures. However, the performance of IVS filters is largely dependent on the statistical dependency measure that is used.

The linear correlation coefficient, R, is a commonly adopted measure of dependence between variables. It forms the basis of such selection schemes as correlation analysis and principal component analysis; both of which have been applied extensively to the development of ANN models of environmental processes (*Olsson et al.*, 2004; *Sousa et al.*, 2007). Two key issues for these (and similar) IVS techniques are the sensitivity of the linear correlation coefficient to noise, and to data transformations during preprocessing, which can influence the apparent relevancy of input variables (*Battiti*, 1994). However, more importantly, the underlying assumption of linearly structured dependence is contradictory to the development of statistical models of non-linear systems.

Mutual information has recently been found to be a more suitable measure of dependence for IVS during ANN development, since it is an arbitrary measure, and makes no assumption regarding the structure of the dependence between variables. It has also been found to be robust due to its insensitivity to noise and data transformations (*Battiti*, 1994; *Darbellay*, 1999; *Soofi and Retzer*, 2003). However, several issues have arisen in the formulation of MI-based IVS algorithms, which are: the ability of the selection algorithm to consider the interdependencies between candidates (redundancy handling); and the lack of an appropriate analytical method for determining when the optimal set has been selected (*Chow and Huang*, 2005). One particular algorithm has been developed that overcomes these difficulties by using the concept of partial mutual information (*Sharma*, 2000). However, it relies upon a computationally intensive bootstrap estimation technique to implement an automatic termination criterion, which necessitates a trade-off between efficiency and the accuracy of selection.

6.2.2 Estimation of partial mutual information

Given a random output variable Y, there will be some uncertainty surrounding an observation $y \in Y$, which can be defined according to the Shannon entropy, H (*Shannon*, 1948). Now, given a random input variable X, which Y is dependent upon, then the mutual observation of (x, y) reduces this uncertainty, since knowledge of x allows inference of the value of y, and *vice versa*. By definition, the mutual information, $I(X;Y)$, is the reduction in uncertainty with respect to Y due to observation of X (*Cover and Thomas*, 1991). This is represented by the intersecting region in Figure 6.1, where the reduced uncertainties surrounding X and Y are denoted by the conditional entropies $H(X|Y)$ and $H(Y|X)$, respectively.

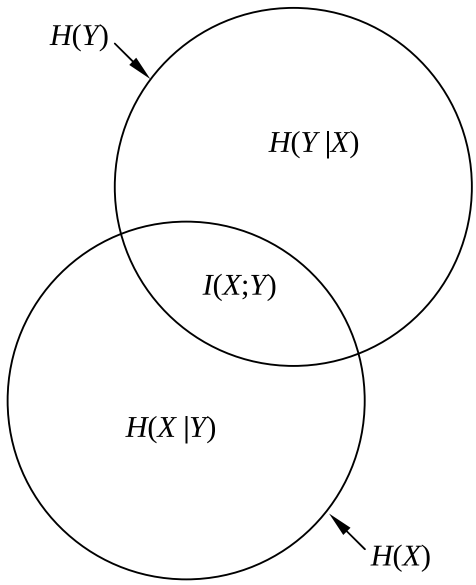


Figure 6.1: Venn diagram representation of the relationship between MI and entropy for output Y and single input variable X.

Mutual information can be determined directly using

$$I(X;Y) = \iint p(x,y) \log \frac{p(x,y)}{p(x)p(y)}\, dxdy, \tag{6.1}$$

where $p(y)$ and $p(x)$ are the marginal probability density functions (pdfs) of X and Y, respectively; and $p(x,y)$ is the joint pdf. However, within a practical context, the true functional forms of the pdfs in (6.1) are typically unknown. Hence, estimates of the densities are used instead. Substitution of density estimates into a numerical approximation of the integral in (6.1) gives

$$I(X;Y) \approx \frac{1}{n} \sum_{i=1}^{n} \log \left[\frac{f(x_i,y_i)}{f(x_i)f(y_i)} \right], \tag{6.2}$$

where f denotes the estimated density based on a sample of n observations of (x,y). Note that the base of the logarithm varies within the literature, and use of either 2 or e is often reported, although the natural logarithm is assumed in this study, unless otherwise stated.

Given the form of (6.2), it follows that efficient and accurate estimation of MI is largely dependent on the technique employed to estimate the marginal and joint pdfs. Non-parametric density estimation techniques are typically considered suitably robust and accurate. In particular *kernel density estimation* (KDE) is used, although it is somewhat computationally intensive compared to alternatives, such as the histogram (*Scott*, 1992). The simple Parzen window forms the basis for this approach, in which an estimator for f is given by

$$\hat{f}\left(x\right) = \frac{1}{n}\sum_{i=1}^{n} K_h\left(x - x_i\right), \tag{6.3}$$

where $\hat{f}(x)$ denotes the estimate of the pdf at x; $x_i\{i = 1, \ldots, n\}$ denote sample observations of X; and K_h is some kernel function (*Scott*, 1992) for which h denotes the kernel bandwidth (or, smoothing parameter). A common choice for K_h is the Gaussian kernel,

$$K_h = \frac{1}{\left(\sqrt{2\pi}h\right)^d \sqrt{|\Sigma|}} \exp\left(\frac{-\left\|x - x_i\right\|}{2h^2}\right). \tag{6.4}$$

Here, d denotes the number of dimensions of X, Σ is the sample covariance matrix, and $\|x - x_i\|$ is the Mahalanobis distance metric, which is given by

$$\|x - x_i\| = (x - x_i)^T \Sigma^{-1} (x - x_i). \tag{6.5}$$

Substituting the expression for the kernel into (6.3), the estimator for f becomes

$$\hat{f}\left(x\right) = \qquad \frac{1}{n\left(\sqrt{2\pi}h\right)^d \sqrt{|\Sigma|}} \sum_{i=1}^{n} \exp\left(\frac{-\left\|x - x_i\right\|}{2h^2}\right). \tag{6.6}$$

Other choices of kernel may be adopted for reasons of computational efficiency (*Bonnlander and Weigend*, 1994), however the performance of the kernel estimator, in terms of accuracy, is dependent more on the choice of bandwidth, than on the choice of kernel itself (*Scott*, 1992).

The optimal choice of bandwidth will depend largely on the distribution of the available data sample. A bandwidth that is too small may be sensitive to noise within the data sample (under-smooth), resulting in a highly variable estimate of MI. On the other hand, a bandwidth that is too large will tend to over-smooth the complex features of the pdf and MI will be underestimated (high bias). Several algorithms, including cross-validation (CV) and plug-in (PI) bandwidth selection, can be used to optimise the kernel bandwidth, but at significant computational expense. *Sharma* (2000), *Bowden et al.* (2005), and *Huang and Chow* (2005) adopted the Gaussian reference bandwidth, h_G for MI estimation as an efficient choice, and *Harrold et al.* (2001) empirically found that a bandwidth of $\sim 1.5h_G$ gave stable estimates of MI. The Gaussian reference bandwidth is determined by the following rule (*Silverman*, 1986)

$$h_G = \left(\frac{1}{d+2}\right)^{1/(d+4)} \sigma n^{-1/(d+4)}, \tag{6.7}$$

where σ is the standard deviation of the data sample. The optimality of h_G for a given set of data might be questionable if the data are not Gaussian, and the Gaussian bandwidth can also tend to over-smooth (*Scott*, 1992). However, it has been noted that the bandwidth can vary by as much as 20% before any degradation in the accuracy of the density estimation becomes noticeable (*Scott*, 1992). Consequently, the reference bandwidth appears to be a reasonable first choice, given its efficiency and widespread use within the literature, and has therefore been adopted in this paper.

The notion of MI is easily extended to systems where the response variable Y is dependent on multiple input variables (*Cover and Thomas*, 1991; *Soofi and Retzer*, 2003). An example of such a system is depicted in Figure 6.2 for the case of two input variables, X and Z. Given X and the already reduced uncertainty $H(Y|X)$ represented in Figure 6.1, the partial mutual information is defined as the further reduction in the uncertainty surrounding Y that is gained by the additional mutual observation of Z.

In fact, PMI is analogous to the partial correlation coefficient, $R'_{ZY \cdot X}$, which quantifies the linear dependence of Y on variable Z that is not accounted for by the input variable X. This is calculated by first filtering both Y and Z via regression on X to obtain residuals u and v, respectively. The Pearson correlation $R\left(u, v\right)$ can then be used to determine $R'_{ZY \cdot X}$. Using this analogy, the PMI can be estimated provided that a suitable regression technique is applied to filter the arbitrary, rather than linear, dependence between variables.

Based on the KDE approach, an estimator for the regression of Y on X is written

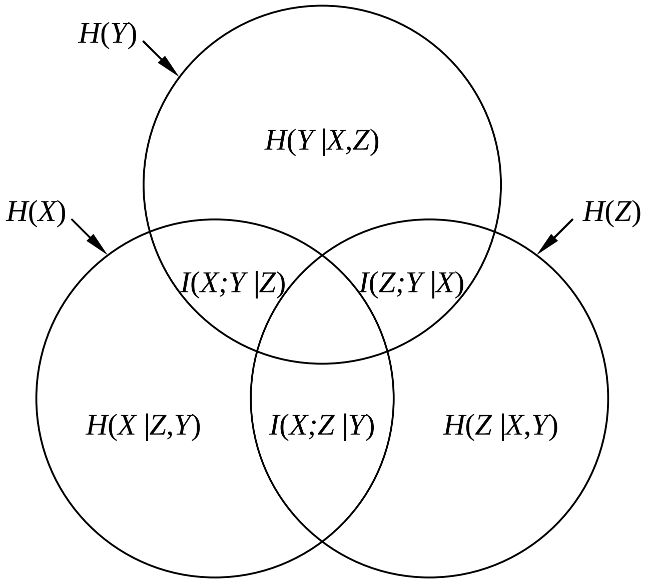


Figure 6.2: Venn diagram representation of the relationship between PMI and entropy for output Y and input variables X and Z.

as

$$\hat{m}_Y(x) = E\left[y|X = x\right] = \frac{1}{n}\frac{\sum_{i=1}^{n} y_i K_h\left(x - x_i\right)}{\sum_{i=1}^{n} K_h\left(x - x_i\right)} \tag{6.8}$$

where $\hat{m}_Y(x)$ denotes the regression estimator; n is the number of observed values (y_i, x_i); K_h is as given in (6.5) and $E[y|X = x]$ denotes the conditional expectation of y given an observed x. An estimator $\hat{m}_Z(x)$ can be similarly constructed, and the residuals u and v can be subsequently obtained using the expressions

$$u = Y - \hat{m}_Y(X) \tag{6.9}$$

and

$$v = Z - \hat{m}_Z(X). \tag{6.10}$$

Using the residuals obtained in (6.9) and (6.10), the PMI is then calculated as

$$I'_{ZY\cdot X} = I(v;u), \tag{6.11}$$

where the subscript notation $I'_{ZY \cdot X}$ is used to denote the PMI, otherwise written as $I\left(Z;Y\ |X\right)$. This notion of PMI allows for the evaluation of the dependence between variables that takes into consideration any information already provided by a given variable X.

6.2.3 Description of the PMIS algorithm

The PMI-based input selection (PMIS) algorithm proposed in this study was originally developed by *Sharma* (2000) for the identification of inputs for hydrological models. Given a candidate set, C, and output variable, Y, the PMIS algorithm proceeds at each iteration by finding the candidate c_s that maximises the PMI with respect to the output variable, conditional on the inputs that have been previously selected. The statistical significance of the PMI estimated for c_s is assessed based on confidence bounds drawn from the distribution generated by a bootstrap loop. If the input is significant, c_s is added to $\mathbf{X}$ and the selection continues; otherwise, there are no more significant candidates remaining and the algorithm is subsequently terminated. The details of the algorithm are as follows:

```
i.    Let $\mathbf{X} \rightarrow \phi$ (Initialisation)
ii.   While $C \neq \phi$ (Forward selection)
iii.    Construct kernel regression estimator $\hat{m}_Y(\mathbf{X})$
iv.     Calculate residual output $u = Y - \hat{m}_Y(\mathbf{X})$
v.      For each $c \in C$
vi.       Construct kernel regression estimator $\hat{m}_c(\mathbf{X})$
vii.      Calculate residual candidate $v = c - \hat{m}_c(\mathbf{X})$
viii.     Estimate $I(v;u)$
ix.     Find candidate $c_s$ (and $v_s$) that maximises $I(v;u)$
x.      For b = 1 to $B$ (Bootstrap)
xi.       Randomly shuffle $v_s$ to obtain $v_s^*$
xii.      Estimate $I_b = I(v_s^*;u)$
xiii.    Find confidence bound $I_b^{(95)}$
xiv.    If $I(v_s,u) > I_b^{(95)}$ (Selection/termination)
```

```
xv.       Move $c_s$ to $\mathbf{X}$
xvi.    Else
xvii.       Break
xviii. Return selected input set $\mathbf{X}$.
```

Here, B is the bootstrap size; and $I_b^{(95)}$ denotes the 95^{th} percentile bootstrap estimate of the randomised PMI, I_b.

The PMIS algorithm is structured in a similar fashion to earlier MI-based IVS algorithms, such as mutual information feature selection (MIFS) (*Battiti*, 1994), but has two advantages. First, PMIS inherently handles redundancy within the candidate set through the direct estimation of PMI, whereas MIFS approximates the effect of selected inputs by means of a heuristic weighting factor. Second, while MIFS uses greedy selection of a pre-specified number of input variables, PMIS includes a criterion that automatically determines the optimum point at which to terminate the selection procedure.

The bootstrap is a statistical tool that is often used to test the quality of statistical estimates based on a finite sample of data (*Hastie et al.*, 2001). Given a statistic $S(x)$, the bootstrap involves drawing B samples $x_1, \ldots, x_B$ from the distribution of x and estimating $S(x_i)$ to determine the influence of the sample on S. The properties of the distribution $p(S)$, such as the mean, variance and percentiles, can be empirically determined based on the B sample estimates $\hat{S}$. In most respects, the bootstrap is equivalent to Monte Carlo simulation and analysis (*Hastie et al.*, 2001). In PMIS, the goal of the bootstrap is to determine an upper bound on the estimate of MI between independent v_s^* and output y. The bootstrap is performed by drawing B independent, uniform random shuffles of v_s, estimating each I_b, and then determining $I_b^{(95)}$. This value represents an approximation to the critical value of I (at a 5% confidence level) for the corresponding sample size, which is used to decide whether the most salient variable at each iteration is statistically relevant, or irrelevant.

The size of the bootstrap, B, is an important consideration in the implementation of PMIS, since it can influence both the accuracy and overall computational efficiency of the algorithm. A large body of literature exists on the bootstrap, which discusses the effect of bootstrap size on the accuracy of confidence bounds for sample-estimates of statistics. The number of bootstraps required can depend on the variability of sample estimates, which will itself depend on the statistic. The quality of data and degree of noise will also determine how large the boostrap size should be to accurately estimate the uncertainty of a given estimate. It has been suggested that a bootstrap size as large as 5 000 may be required for a suitably reliable estimate (*Chernick*, 1999).

In the case of the PMIS algorithm, a bootstrap size of 100 has been used previously to estimate confidence bounds on the error in MI estimates, for reasons of computational efficiency (*Sharma*, 2000; *Bowden et al.*, 2005). However, a bootstrap this small might not provide reliable estimation of the confidence bound, which could result in unreliable and/or sub-optimal input variable set selection. The reliability of PMIS with 100 bootstraps has not yet been ascertained, nor has a reliable bootstrap size been determined for this application. However, given the computational requirement for a single estimate of MI, any significant increase in the bootstrap size beyond 100 is undesirable, as the computation time required to implement PMIS would become excessive. Hence, what is needed is either an alternative to the bootstrap, which yields a more accurate estimate of the MI error without increasing the computational effort required, or possibly a termination criterion that is not based upon a direct comparison with the critical value of MI.

6.3 Formulation of Alternative Termination Criteria

The remainder of this paper describes an assessment of the sensitivity of the existing PMIS termination criterion to the size of the bootstrap, and the novel formulation of several alternative termination criteria. The motivation for formulating a suitable alternative to the existing termination criterion was to improve the selection accuracy and overall computational efficiency of the PMIS algorithm. In total, three new termination criteria are proposed that each overcome the limitations of the bootstrap in the existing algorithm.

6.3.1 Modified bootstrap

The performance of the bootstrap, in terms of accuracy and computational requirement, is largely observed to be a function of the bootstrap size, B, and of the confidence bound selected. The original implementation adopted $B = 100$ and $I_b^{(95)}$ (*Sharma*, 2000), however, there has been relatively little examination of the performance of the PMIS algorithm using either a larger bootstrap size, or a stricter confidence bound. Consequently, an investigation into the performance of the PMIS algorithm with modified bootstrap parameters was undertaken to provide some benchmark results of the efficiency and accuracy of the bootstrap-based approach.

In this study, a confidence level of $I_b^{(99)}$ was trialled to compare with the original $I_b^{(95)}$ confidence bound. It was expected that a more strict confidence bound could potentially reduce the degree of over-specification because it was a more difficult test to pass. An increased bootstrap size of 1 000 was investigated to assess the potential improvement in the accuracy of estimations of both $I_b^{(95)}$ and $I_b^{(99)}$ this would provide. A ten-fold increase in bootstraps was considered a suitable increase to gauge the effect of bootstrap size on accuracy, while maintaining reasonable analysis times. The performance of PMIS using the increased bootstrap size provided a useful benchmark for a more accurate bootstrap-based termination criterion, against which the accuracy and reliability of alternative criteria investigated in this paper could be compared.

6.3.2 Tabulated critical values

Tables of the critical values of the correlation coefficient, R, are readily available, which are based on the analytical formula for the distribution of the error of an estimate for a given sample size. In the case of the linear correlation coefficient, R, the distribution of a sample-estimate follows a t-distribution. Based on the t-distribution, tables of the critical value of the correlation coefficient, R, are easily constructed, as in *David* (1966), which provide the critical value of R for the number of samples, and a given confidence level. However, unlike the linear correlation coefficient, an equivalent analytical expression for I cannot be derived due to the form of the expression in (6.2) (*Goebel et al.*, 2005). Hence, practitioners must resort to bootstrapping in order to estimate $f(\hat{I})$ (such as in *Granger et al.* (2004) and *Sharma* (2000)). However, a recent study undertaken by *Granger et al.* (2004), in which the distribution of a kernel-based information estimator $\hat{s}_p$ was examined for a number of time-series models, suggests a practical alternative to the bootstrap.

Instead of using analytical values, a method for constructing tables of estimated critical values of I using Monte Carlo simulation is described by *May et al.* (2006). Monte Carlo simulation was used to empirically determine the distribution for the MI estimator described in Section 6.2 as the first step in developing a termination criterion based on approximate critical values. In each simulation, the MI was estimated for a dataset comprising i.i.d Gaussian white-noise data, with sample size n ranging from 50 to 5000 samples, in order to obtain a set of critical values that could be used for testing for independence based on MI. For each sample size, a series $\varepsilon_y \sim N(0,1)$ was generated first and the marginal pdf f_{ε_y} estimated.

A total of 100 000 independent replicates of series $\varepsilon_x \sim N(0,1)$ were generated, independent of ε_y. For each instance of ε_x the pdfs f_{ε_x} and $f_{\varepsilon_x \varepsilon_y}$ were estimated and $\hat{I}(\varepsilon_x, \varepsilon_y)$ was subsequently evaluated.

The resulting critical values of I are given in Table 6.1 for different confidence levels. Two alternative termination criteria were formulated whereby, at each iteration, the estimated $I'_{C_s Y \cdot S}$ is compared to the respective critical values $I^{(95)}$ and $I^{(99)}$ obtained from Table 6.1, rather than those directly estimated by the bootstrap, in order to decide whether the candidate variable should be selected, or the algorithm terminated. The elimination of the computationally expensive bootstrap loop resulted in a much faster overall IVS procedure.

6.3.3 AIC-based criterion

An alternative termination criterion was formulated in this study that was based on analysis of the output variable residual, u, that results from regression of Y on the newly formed set, X. This criterion is based on the assumption that, as the optimal set S is constructed incrementally during successive iterations of the forward selection loop, the non-parametric regression $\hat{m}_Y(X)$ will increasingly filter more of the information contained within Y. Eventually, when the optimal input variable set is reached, the kernel regression will show no further reduction in the information contained in u, and selection is terminated.

The Akaike information criterion (AIC) (*Akaike*, 1974) was adopted as a measure of the trade-off between accuracy of the regression filter and the size of the input set X, for the purposes of formulating this termination criterion. Measures such as the AIC are commonly used as a basis for comparison in model selection. The AIC is given as

$$\mathrm{AIC} = n \log_e \left(\frac{1}{n} \sum_{i=1}^{n} u_i^2 \right) + 2p, \tag{6.12}$$

where n is the number of observations, u_i denote n residuals, and p is the number of model parameters. In the case of linear regression, the term p is equal to $k+1$, where k is the number of variables. However, for non-parametric regression, it is necessary to use a measure of complexity such as the *effective number of parameters*, or the Vapnik-Chernovekis (VC) dimension. The effective number of parameters, d can be determined by trace(S), where S is the $n \times n$ *hat-matrix*

Table 6.1: Critical values of the KDE-based mutual information estimator (after *May et al.* (2006)).

n	$\bar{I}$	$I^{(90)}$	$I(95)$	$I^{(99)}$
50	0.1323	0.1990	0.2224	0.2705
60	0.1236	0.1825	0.2031	0.2452
70	0.1166	0.1694	0.1879	0.2254
80	0.1106	0.1592	0.1756	0.2091
90	0.1057	0.1506	0.1657	0.1973
100	0.1013	0.1429	0.1572	0.1858
120	0.0943	0.1309	0.1434	0.1688
140	0.0883	0.1211	0.1321	0.1546
160	0.0839	0.1138	0.1237	0.1444
180	0.0798	0.1072	0.1166	0.1356
200	0.0763	0.1019	0.1103	0.1276
220	0.0735	0.0975	0.1055	0.1215
240	0.0707	0.0932	0.1005	0.1158
260	0.0682	0.0894	0.0965	0.1108
280	0.0661	0.0862	0.0928	0.1062
300	0.0642	0.0834	0.0896	0.1022
400	0.0567	0.0724	0.0775	0.0876
500	0.0513	0.0646	0.0689	0.0775
600	0.0473	0.0589	0.0627	0.0702
700	0.0441	0.0544	0.0578	0.0644
800	0.0415	0.0509	0.0539	0.0597
900	0.0393	0.0479	0.0507	0.0563
1000	0.0375	0.0455	0.0481	0.0531
2000	0.0270	0.0318	0.0333	0.0361
3000	0.0222	0.0257	0.0268	0.0289
4000	0.0192	0.0221	0.0230	0.0247
5000	0.0172	0.0196	0.0204	0.0218

defined by the expression

$$\hat{y} = Sy, \tag{6.13}$$

where the elements S_{ij} correspond to $K_h(x_i, x_j)$. It would be expected that different complexity measures (such as AIC, Bayesian information criterion (BIC), minimum description length (MDL)), which each penalise model complexity differently, could potentially result in different selections. However, comparisons of model selection criteria are beyond the scope of this paper, and the AIC was therefore adopted.

During PMIS, the behaviour of the AIC will initially be dominated by a reduction in the magnitude of the residual terms, and decreases with increasing k before reaching some minimum value. Beyond this point, the AIC increases due to the $2p$ term, which penalises the selection of additional variables. Hence, the optimal value of k corresponds to the minimum AIC. Using this termination criterion, the PMIS algorithm no longer includes the bootstrap, nor is the PMI compared to a critical value. Rather, the AIC is determined for the input set that includes C_s and if the AIC decreases, then C_s is selected; otherwise C_s is rejected and the forward selection procedure is terminated.

6.3.4 Hampel test criterion

Outlier detection methods are a robust statistical approach for determining if a given value, x, is significantly different from others within a set of values X. Outlier detection is commonly used to identify interesting data for further analysis, or remove spurious data prior to analysis. In the case of PMIS, having identified the most relevant candidate at each round using the outlier test, the decision can subsequently be made to either select such a candidate and continue if the candidate is classified as an outlier (i.e. the PMI is significantly higher than all others), or terminate selection if the candidate is not classified as an outlier, that is, it is not significant. This forms the basis for the third proposed termination criterion.

The Z-test is commonly adopted for the detection of outliers within a given population of observed values of a given variable. The test compares the deviation of a single observation from the sample mean of all observations. An observed value with a Z-score greater than three is typically considered to be an outlier based on the 3σ rule for Gaussian distributions (i.e. outliers lie greater than three standard deviations from the population mean). However, the presence of multiple outliers within a population can significantly alter the robustness of this test. The reason for this is that the mean is sensitive to the number of outliers within the population, since just one very distant outlier can significantly increase the mean and variance sufficiently to make the outliers seem less different, relative to the whole population. The effect is referred to as *masking* since this effectively hides outliers. In statistical literature, the sensitivity of outlier detection methods to masking is determined by the breakdown point of the test, which simply refers to the proportion of outliers that must be present to significantly alter the location (i.e. mean) and spread (i.e. variance) of the population. The breakdown point of the Z-test is $1/n$, since only one sufficiently large outlier will cause the test to breakdown (*Davies and Gather*, 1993).

In formulating an outlier detection-based termination criterion for PMIS, the underlying assumption is that a set of candidates will initially contain some proportion of redundant and irrelevant variables, and significant variables will be detected. However, potential masking of outliers was an important consideration, given that the candidate set is likely to contain more than one relevant variable. Hence, a modified Z-score, which utilises the *Hampel distance*, was adopted instead to increase the robustness of the approach. The Hampel distance (*Davies and Gather*, 1993) is based upon the population median. The breakdown point of the Hampel-test is $n/2$, and is considered to be one of the most robust outlier tests in the presence of multiple outliers (*Davies and Gather*, 1993; *Pearson*, 2002). The Hampel test begins by calculating the absolute deviation from the median PMI for all candidates according to

$$d_j = \left| I_{X_jY\cdot S} - I^{(50)}_{X_jY\cdot S} \right|, \tag{6.14}$$

where d_j denotes the absolute deviation; and $I^{(50)}_{X_jY\cdot S}$ denotes the median PMI for candidate set C. The Hampel distance can then be determined by

$$Z_j = \frac{d_j}{1.4826 d_j^{(50)}}, \tag{6.15}$$

where Z_j denotes the Hampel distance (modified Z-score) for candidate X_j; and $d_j^{(50)}$ denotes the median absolute deviation (MAD), d_j. The factor of 1.4826 scales the distance such that the rule $Z > 3$ can be applied, as is the case for the conventional Z-test (*Pearson*, 2002).

Using this termination criterion, the PMIS algorithm again no longer includes the bootstrap loop, nor is the PMI compared to any critical value of I. Instead, the value Z_s is determined for candidate C_s and if $Z_s > 3$, the candidate is selected and added to S; otherwise the forward selection algorithm is terminated.

6.4 Experimental Methods

The task of selecting the correct input variables for a suite of benchmark data-generating processes formed the basis of evaluation of the alternative termination criteria, which is consistent with previous studies, where synthetic datasets were used to test the performance of novel IVS algorithms (*Chow and Huang*, 2005; *Huang and Chow*, 2005; *Bowden et al.*, 2005; *Sharma*, 2000; *Battiti*, 1994; *Kwak and Choi*, 2002; *Bonnlander and Weigend*, 1994). The datasets generated in this study, although synthetic, are typical of environmental modelling applications in that they represent a range of time-series and input-output functions of varying degrees of non-linearity and persistence. Use of synthetic data for benchmarking and analysis is useful since a comparison can be made between the specified variables and the known set of “true” input variables. Furthermore, features of synthetically generated data, such as signal-to-noise ratio, dimensionality, and sample size can be adjusted to allow for a more comprehensive analysis of the factors that influence IVS techniques. The application of PCIS and PMIS to both linear and non-linear data serves as a basis for comparison of the relative ability of the dependence measures considered (R and I, respectively) to identify relevant relationships.

The models shown in Table 6.2 were used to generate data sets of varying sample length, linearity, and noise, where in all cases the term $\varepsilon \sim N(0,1)$ denotes the additive noise. The suite of models comprises a mixture of time-series models, for which the output x_t is a function of the set of past observations $\{x_{t-1},\ldots,x_{t-d}\}$; and input-output models, where the output variable y is represented by some transfer function $f(x_1,\ldots,x_d)$. These models therefore represent test cases for such data-driven applications as time-series forecasting and function approximation.

Models AR4 and AR9 denote linear auto-regressive time-series models (fourth and ninth order, respectively). TAR1 and TAR2 are both non-linear threshold auto-regressive time-series models. Data generated by these four models have been used previously (*Sharma*, 2000). Datasets for the time-series models were generated with 15 lagged observations (i.e. $d = 15$) to form the candidate set. The Friedman model is a five-dimensional input-output function that is recommended for benchmarking non-linear regression (*Friedman*, 1988). The Fried-

Table 6.2: Benchmark data-generating models.

Linear auto-regressive time-series		
1.	AR4	$x_t = 0.6x_{t-1} - 0.4x_{t-4} + \varepsilon_t$
2.	AR9	$x_t = 0.3x_{t-1} - 0.6x_{t-4} - 0.5x_{t-9} + \varepsilon_t$
Non-linear threshold auto-regressive time-series		
3.	TAR1	$x_t = \begin{cases} -0.9x_{t-3} + 0.1\varepsilon_t & \text{if } x_{t-3} \leq 0, \\ 0.4x_{t-3} + 0.1\varepsilon_t & \text{otherwise.} \end{cases}$
4.	TAR2	$x_t = \begin{cases} -0.5x_{t-6} + 0.5x_{t-10} + 0.1\varepsilon_t & \text{if } x_{t-6} \leq 0, \\ 0.8x_{t-10} + 0.1\varepsilon_t & \text{otherwise.} \end{cases}$
Non-linear input-output functions		
5.	Friedman	$y = 5\left(2\sin(\pi x_1 x_2) + 4(x_3 - 0.5)^2 + 2x_4 + x_5\right) + \varepsilon$
6.	Mexican Hat	$y = \frac{\sin\left(\sqrt{x_1^2+x_2^2}\right)}{\sqrt{x_1^2+x_2^2}} + \varepsilon$

man dataset nominally includes five additional noise variables. However, a modified form of the Friedman dataset was also considered, in which an additional five dimensions of noise variables (ten in total) were added to the input variables to form the candidate set. This was to investigate the sensitivity of the Hampel test criterion to the relative proportion of irrelevant candidate variables. The Mexican Hat function, which is a well-known two-dimensional non-linear function, provided a third input-output case. Data were generated for the Mexican Hat model with an additional 13 noise variables to create a total of 15 candidate input variables.

In this study, 30 independent instances of each model were used to generate data sets of 50, 100, 500, and 1 000 observations of $X(t)$ (or, (X, y)). Using each of the termination criteria described in Section 6.3, IVS algorithms were applied to select the input variables for each model. For the purpose of this study, all models were implemented as C++ classes, which allowed the instantiation of models at run-time to simultaneously generate multiple data sets, as required. Time-series models were initialised with the initial conditions of $x_0 \sim N(0, 1)$ and $x_t = 0 \; \{\forall t < 0\}$, and the first 20 observations were discarded to avoid initialisation effects. In the case of the input-output functions, each of the candidate variables were uncorrelated Gaussian $\sim N(0,1)$ noise.

6.4.1 Comparison to IVS based on the correlation coefficient

The implementation of MI filters comes at considerable computational effort compared with correlation-based filters, due to the effort involved in pdf estimation. Although *Sharma* (2000) gives possible examples where correlation will fail, the overall performance of correlation and MI is yet to be thoroughly examined. Consequently, a selection scheme, based on partial correlation, serves as a basis for comparison between the linear and non-linear approaches to IVS. The algorithm, hereafter referred to as partial correlation input selection (PCIS), is structured as for PMIS, but with the linear correlation coefficient used to measure the relevance of the candidates. Furthermore, generalised linear regression (GLR) is used to estimate the residuals rather than kernel regression. GLR describes a linear filter of the form

$$y = \mathbf{B}\mathbf{x} + \varepsilon, \tag{6.16}$$

where $\mathbf{B}$ is a vector of linear regression coefficients and ε is the residual noise. The least-squares estimate of the coefficients $\mathbf{B}$, for a sample (X, Y), can be determined as

$$\mathbf{B} = (\mathbf{X^T X})^{-\mathbf{1}}\mathbf{X^T Y}. \tag{6.17}$$

Based on the original implementation of PMIS, the linear PCIS algorithm proceeds as follows:

```
i.   Let $X \rightarrow \phi$ (Initialisation)
ii.      While $C \neq \phi$ (Forward selection)
iii.     Construct GLR estimator $\hat{m}_Y(X)$
iv.      Calculate residual output $u = Y - \hat{m}_Y(X)$
v.       For each $c \in C$
vi.          Construct GLR estimator $\hat{m}_c(X)$
```

```
        Calculate residual candidate $v = c - \hat{m}_c(X)$
        Estimate $R(v; u)$
    Find candidate $c_s$ (and $v_s$) that maximises $|R(v; u)|$
    For b = 1 to $B$ (Bootstrap)
        Randomly shuffle $v_s$ to obtain $v_s^*$
        Estimate $R_b = R(v_s^*; u)$
    Find confidence bound $R_b^{(95)}$
    If $R(v_s, u) > R_b^{(95)}$ (Selection/termination)
        Move $c_s$ to $X$
    Else
        Break
Return selected input set $X$
```

Similarly to PMIS, alternative termination criteria were implemented for PCIS. Critical values of the linear correlation coefficient were obtained from tables in *David* (1966) to implement the critical value-based termination criteria.

6.5 Results and Discussion

Figure 6.3(a), which plots the estimated $I'_{C_jY \cdot S}$ and the $I_b^{(95)}$ confidence bound estimated from 100 bootstraps, illustrates the use of the bootstrap-based termination criteria. Here, the forward selection, based on estimation of PMI, correctly selects the "true" input variables for the AR9 model, in their order of relative importance: $x(t-4)$, $x(t-9)$ and $x(t-1)$. However, due to under-estimation of the confidence bound, the estimated value of $I'_{C_jY \cdot S}$ does not satisfy the termination criterion until the eighth iteration, thus resulting in the selection of an additional five variables. A better result was observed for the I_b^{99}, which is shown in Figure 6.3(b). Due to the under sampling of the tails of the error distribution, there is only a small difference between the estimated I_b^{95} and I_b^{99}, but this was sufficient to achieve termination at the correct number of input variables. Use of the $I^{(95)}$ and $I^{(99)}$ confidence bounds obtained from Table 6.1 (shown in Figure 6.3(c) and 6.3(d), respectively), which were more accurately estimated, resulted in the correct selection of input variables. This result highlights the potential sensitivity of the bootstrap-based criteria to bootstrap size, and demonstrates the potentially unreliable nature of PMIS when an insufficient number of bootstraps is used to implement the termination criterion.

Figure 6.3(e) illustrates the use of the AIC-based termination criterion. The minimum AIC of 215 is clearly obtained following the selection of the third input, and subsequently the algorithm terminates at this point, having correctly selected the required inputs. Application of the Hampel distance-based termination criterion for the AR9 example is shown in Figure 6.3(f). The Hampel distance falls below three during the fourth iteration of the algorithm, and hence also terminates at the optimal point, having rejected the most relevant candidate at the fourth iteration.

6.5.1 Selection Accuracy

The application and assessment of the criteria, as shown in Figure 6.3, was repeated for 30 datasets independently generated by each of the models, and for each sample size. The overall results of the benchmarking study are summarised in Figure 6.4–6.7, which show the average selection accuracy achieved on datasets generated by linear models (Figure 6.4 and 6.5) and non-linear models (Figure 6.6 and 6.7) by PMIS and PCIS, respectively, on datasets of length 50, 100, 500 and 1 000.

Each graph indicates the average frequency of under-specified, correct, and over-specified models for each of the two groups of models (linear or non-linear) that was achieved by each algorithm (PMIS and PCIS) using the different termination criteria (A–H). Under-specification was defined as an incomplete set of inputs, and therefore represented a failure of the algorithm to identify all of the relevant variables. Correct specification meant that only the exact input variables for each of the respective functions were selected. Over-specification of the model was defined as the selection of either irrelevant, or redundant, variables in addition to the required input variable set. For each model, the frequency was determined from the selections obtained for the 30 independent tests. A summary of the alternative termination criteria is provided in Table 6.3.

A clear trend in the performance of PMIS when using the different termination criteria was observed, as indicated in Figure 6.5(d) and Figure 6.7(d). Neglecting differences in performance due to sample size, which is discussed later in Section 6.5.4, the results for a sample size of 1 000 typify the relative performance of each algorithm. It was found that the least accurate selections were obtained for criteria A–D, which were based on bootstrap estimates of confidence bounds.

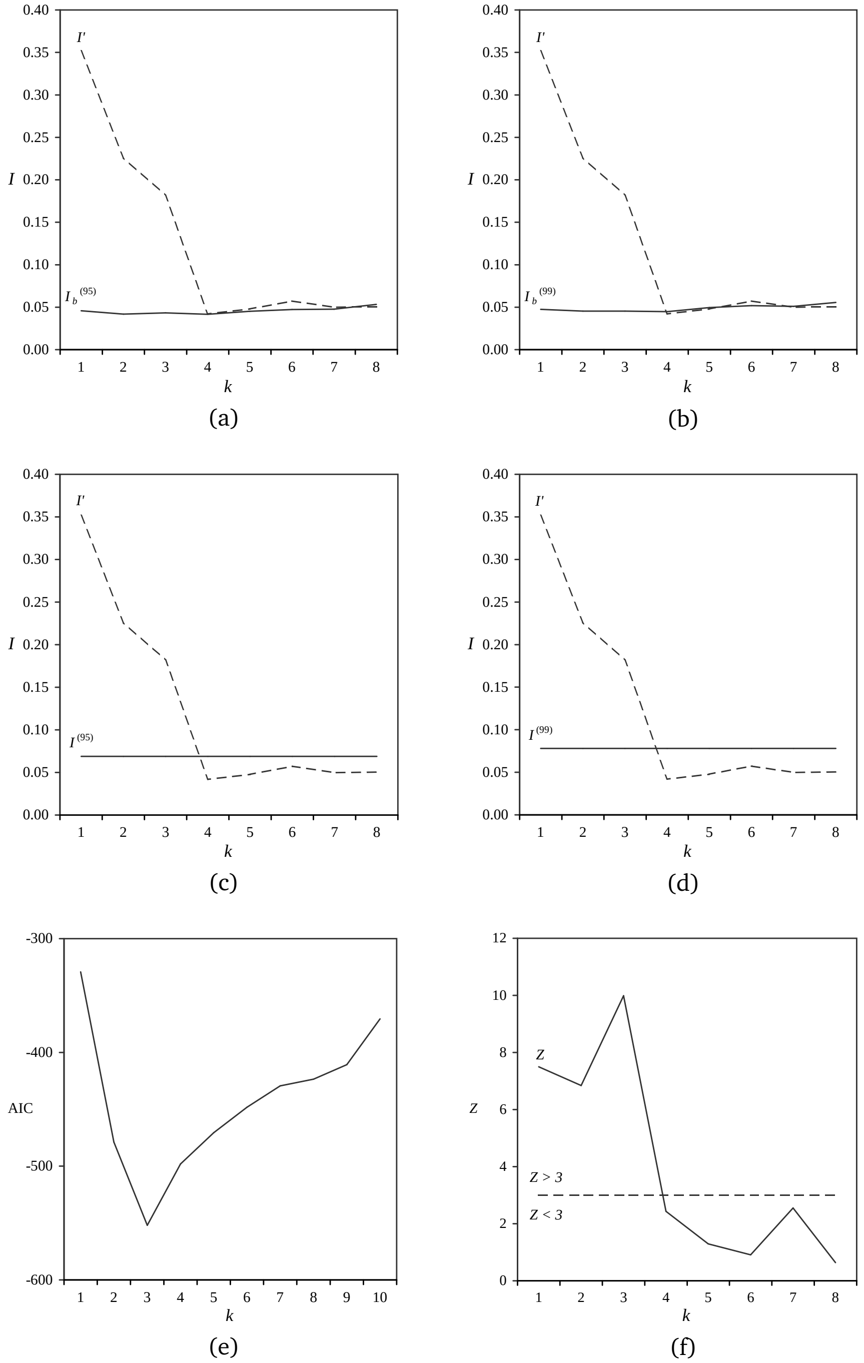
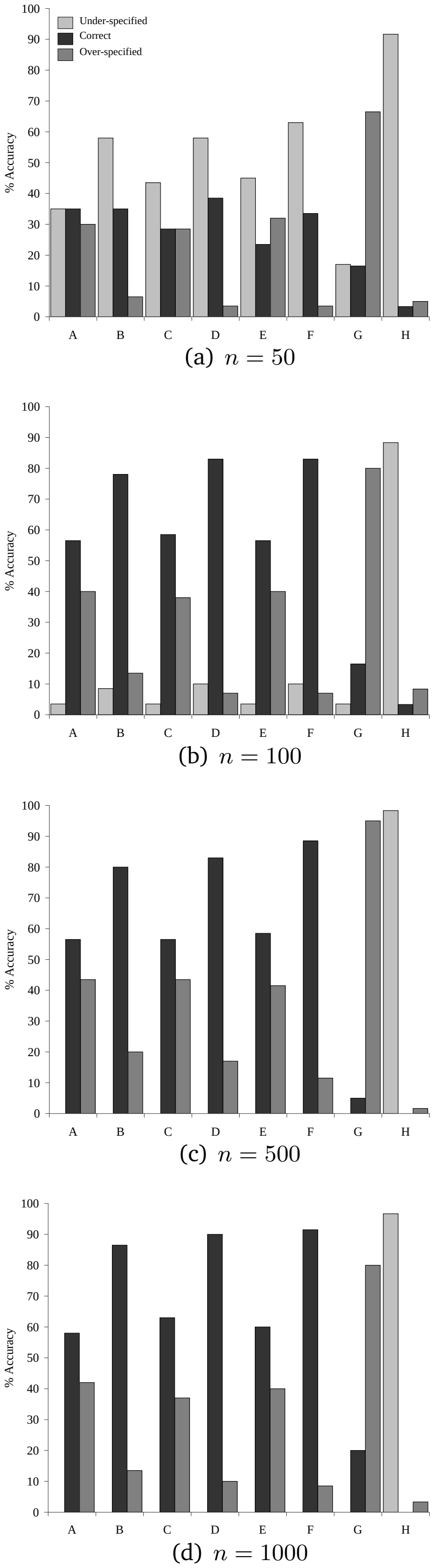


Figure 6.3: Application of alternative PMIS termination criteria for the AR9 time-series example.



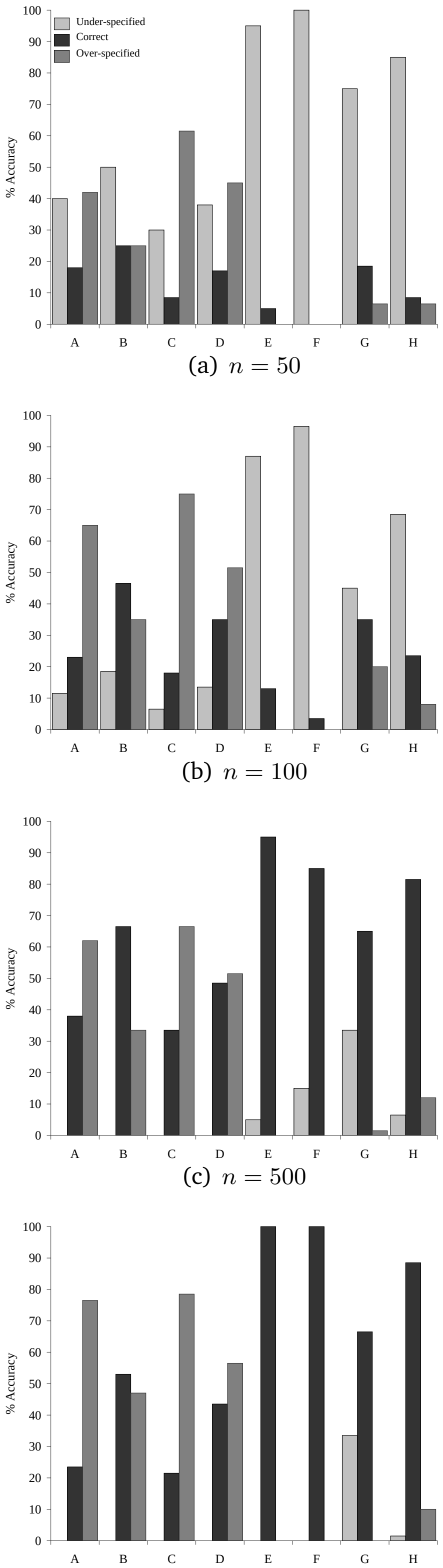
(a) $n = 50$

(b) $n = 100$

(c) $n = 500$

(d) $n = 1000$

Figure 6.4: Overall performance of PCIS for linear datasets



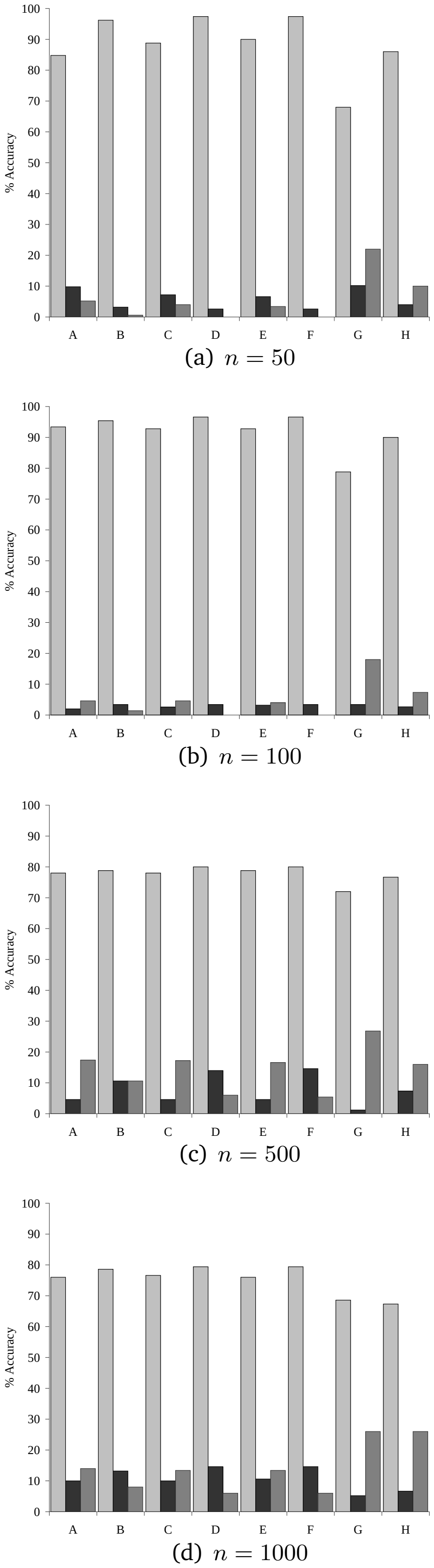
(a) $n = 50$

(b) $n = 100$

(c) $n = 500$

(d) $n = 1000$

Figure 6.5: Overall performance of PMIS for linear datasets



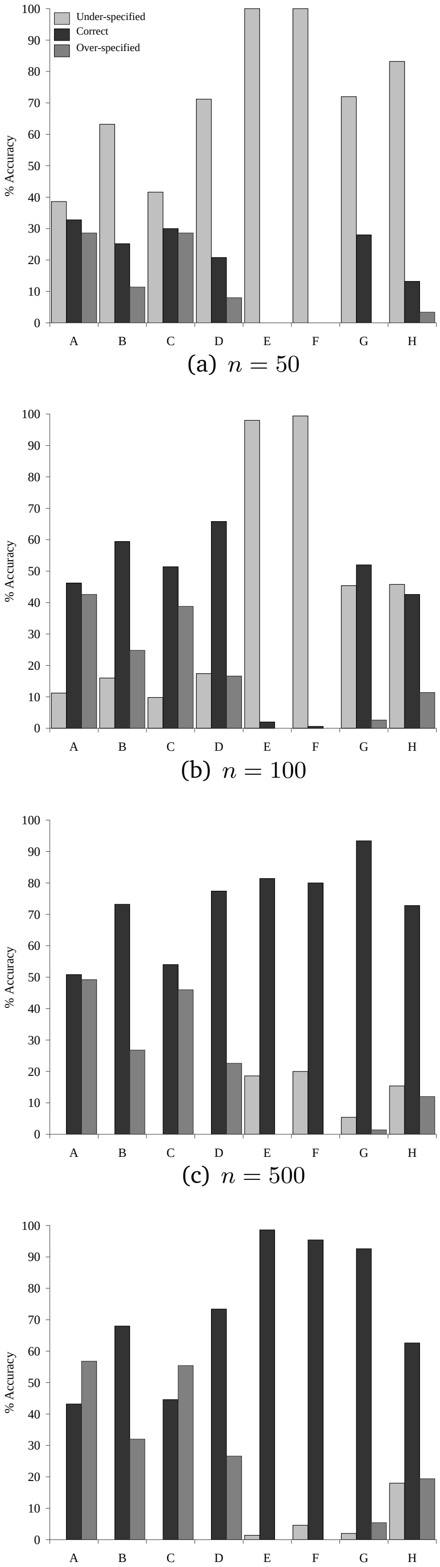
(a) $n = 50$

(b) $n = 100$

(c) $n = 500$

(d) $n = 1000$

Figure 6.6: Overall performance of PCIS for non-linear datasets



(a) $n = 50$

(b) $n = 100$

(c) $n = 500$

(d) $n = 1000$

Figure 6.7: Overall performance of PMIS for non-linear datasets

Table 6.3: Summary of termination criteria assessed in benchmarking study.

Criterion	Description	Notes
A	$I' > I_b^{(95)}$	Small bootstrap estimate, $B = 100$.
B	$I' > I_b^{(99)}$	R' for PCIS
C	$I' > I_b^{(95)}$	Large bootstrap estimate, $B = 1\ 000$.
D	$I' > I_b^{(99)}$	
E	$I' > I^{(95)}$	Critical I taken from Table 6.1 (from *David* (1966) for R).
F	$I' > I^{(99)}$	
G	Minimum AIC(k)	
H	$Z > 3$	

In particular, the use of criteria A and C resulted in comparably poor selections, and showed a tendency to over-specify the input variable set. This showed that the confidence bound was consistently underestimated in these cases, and that a bootstrap size of 100 was insufficient. Increasing the bootstrap size to 1 000 reduced the frequency of over-specification. However, the improvement was only marginal (5–10%), given the increased computational effort incurred. This result suggests that the under-estimation of confidence bounds has a significant influence on the performance of these criteria and that irrespective of the data, too few bootstraps will most likely lead to over-specification. In terms of the choice of confidence bound, the use of $I_b^{(99)}$ resulted in more accurate selections (see criteria B and D), which indicates that use of the higher value confidence bound was able to partially compensate for the under-estimation.

The criteria based on tabulated critical values of MI (E and F) resulted in the best performance with 95–98% selection accuracy overall, with correct selection of 100% of input variable sets for some individual cases. There was no significant difference in performance between use of $I^{(95)}$ and $I^{(99)}$, and both were considered to be appropriate termination criteria based on these results. However, it should be noted that use of the tabulated critical values inherently makes an assumption regarding the distribution of the data (*May et al.*, 2006). In this case, the criteria performed well because the assumed Gaussian distribution matches the data closely, but deviations from the assumed distribution may impact on the appropriateness of the tabulated values. Comparisons between different confidence bounds also show a different trend to those estimated by the bootstrap. By definition, a smaller significance level (higher upper bound) tolerates less uncertainty in the estimation of I. In particular, at smaller sample sizes, where there is greater uncertainty in estimates of I, and a smaller signal-to-noise ratio, the apparent relevance is lower and use of the I^{99} threshold typically selected fewer variables than I^{95}. This trend would be expected to be observed in real-world data, with similarly few data or a high degree of noise.

The results in Figure 6.7(d) show that the AIC-based termination criterion (G) achieved selection accuracy that was comparable with that achieved by using the tabulated critical values (criteria E and F) for the non-linear datasets, indicating that this is also a suitable termination criterion. *Sharma* (2000) observed that the estimation of residuals is only approximate, given that the kernel bandwidth may be sub-optimal, and this may possibly have contributed to the relatively poor performance observed for the linear datasets when this termination criterion was used.

An interesting result was the relatively poor performance of PMIS when the criterion based on the Hampel test was used (criterion H), which resulted in only 63% cases of correct specification overall for the non-linear datasets in Figure 6.7(d). However, the Hampel outlier test is significantly influenced by the proportion of irrelevant variables within the candidate set due to its $n/2$ breakdown point. In the case of the Friedman model, with only five noise variables, the criterion failed to identify the complete set of inputs to the models. Since in this case the relative proportion of noise variables was only 50%, this poor performance was attributed to the masking effect described in Section 6.3.4. In contrast, when the number of noise dimensions was increased to ten (67%), the use of the same termination criterion yielded excellent performance, with PMIS achieving 97% correct specification (for $n = 1\ 000$). The results correspond to the value of the Hampel distance determined for the most salient candidate during the first round of selection. In the case where the model was underspecified, the value was less than 3, which terminated selection with zero input variables selected.

The results obtained for PMIS demonstrate the sensitivity of the Hampel outlier criterion to potential masking effects caused by the presence of more than one relevant candidate. The sensitivity is dependent on the relative proportion of irrelevant variables, since outliers are determined by comparison to a large population of irrelevant candidates. If the proportion of irrelevant candidates is sufficiently small (i.e. proportion of relevant candidates is large), then the Hampel test will potentially be affected by masking and become less able to identify significantly relevant candidates as outliers. The risk of masking is minimised, since the Hampel test has the highest breakdown point of any statistical outlier detection test, and therefore this termination criterion is as robust as can be expected. However, it is conceded that in some cases the issue of masking may be unavoidable, and must be considered when implementing PMIS using the Hampel-test termination criterion.

One solution to the problem has been implicitly presented, in that it is possible to overcome the issue of masking by simply adding noise variables to the candidate set. The addition of “dummy” noise variables can improve the stability of the Hampel test over successive iterations, as the influence of outliers (highly relevant candidates) on the median relevance of all candidates is reduced, and this is evident from the improved performance when the number of noise variables was increased in the experimental Friedman dataset. The problem with such an approach is that the exact number of dummy noise variables that should be added to the candidate set will usually be unknown, since the number of salient variables would need to be known first. It can at least be said that adding $d + 1$ dummy noise variables to the candidates will guarantee that more than 50% of the candidates are irrelevant variables. The drawback with seeding the candidates with so many irrelevant variables is that the number of PMI estimates (and hence, computational effort) that is required will potentially be significantly greater (i.e. double) than for other criteria, such as the AIC-based criterion (G) and the Monte Carlo-derived critical values of I' (E and F).

Recall that, in Section 6.3.2, Monte Carlo estimates of MI for synthetically generated noise variables were generated to derive confidence bounds for the error in MI estimation as a function of sample size. Consequently, an alternative to seeding the candidate set with dummy noise variables to improve the reliability of the Hampel test would be to add MI scores already determined for noise variables, rather than re-estimating them during selection. By using this approach, the median value of the distribution of MI scores for candidates could be made sufficiently stable to avoid any masking effects that might be caused by a high number of relevant candidates, but without requiring any additional computation. This approach essentially follows the same rationale as replacing the bootstrap with the Monte Carlo derived confidence bounds. In fact, if a large enough number of MI estimates for noise variables were added to those of the candidates, it would be expected that the distribution of candidate scores would simply be that of the noise variables, since the MI estimates for the relatively small number of candidates would have less of an influence on the overall distribution. In this case, the only difference between the Hampel and critical Monte Carlo tests would be the method for determining the critical value by analysing the location and spread of the data. The $I^{(90)}$ and $I^{(99)}$ noise thresholds represent an MI that is 3 and 4 standard deviations from the mean, respectively (assuming normally distributed data), while the Z-test is based on the scaled MAD, which is equivalent to 3 standard deviations from the mean.

Wilcox (2001) provides a number of suitable approaches to refining estimates of the median and MAD in the presence of outliers. One simple method for robustifying outlier detection is to use a *clean* population to estimate the distribution parameters (i.e. the median and MAD). By definition, a clean population is free of outliers, and is therefore more indicative of the true properties of the distribution of data, without the undesirable influence of outliers. A clean population may be derived by removing 5% of the data that potentially correspond to outliers, which is an approach that is widely used (*Pearson*, 2002). In PMIS, this would correspond to eliminating the top 5% scoring candidates from the computation of the median PMI and the MAD. This approach seems easy and efficient to implement, and can be applied to any dataset. Alternatively, other methods described by *Wilcox* (2001), such as M-estimators, would be equally as useful in improving the Hampel test. Determining if the population needs to be cleaned can also be based on analysis of selections without cleaning, since masking will result in early termination at the first round of selection.

The influence of masking on the overall performance of PMIS is exaggerated in the specific case of the Friedman model, as it is a synthetic example that represents the worst case, in terms of masking. It is expected that in most real-world applications, where there are usually many variables to consider, it will be unlikely that the issue of masking will be as signficant as in this particular synthetic example. Based on the results for the other datasets, the Hampel outlier test performs efficiently and is likely to work well in most applications. Consequently, the PMIS algorithm with the Hampel-test criterion will be expected to perform well for most real-world datasets. This is particularly true for time-series applications, where the number of candidate inputs is likely to be large as a result of the inclusion of lagged variables. It is expected that with a sufficient number of lags, there will be a majority of variables that are irrelevant after the first round of selection, and more that become irrelevent (due to redundancy) as selections are made. Note that the PMIS algorithm is easily modified to select the candidate with the highest MI during the first round of selection, and then apply the termination criterion to all subsequent rounds, which can minimise masking when selecting lags.. However, in the event that masking remains an issue, diagnosing the problem is straightforward, since no selections will be made, and the Hampel test can be easily modified to address the issue, as discussed above.

6.5.2 Computational efficiency

In addition to improved accuracy of PMIS, in comparison with the original implementation of the algorithm, the use of criteria E, F, G, and H also achieved a significant improvement in the overall computational efficiency of the forward selection loop. This was specifically due to the elimination of the requirement for the computationally expensive bootstrap. The proportion of computational effort involved in the implementation of the bootstrap can be estimated on the basis that the $O\{n^2\}$ estimation of PMI dominates the computation. Given the

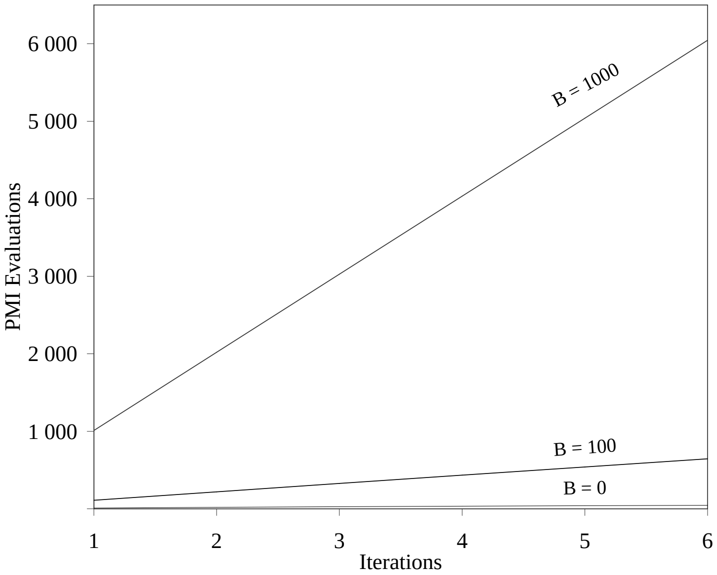


Figure 6.8: Comparison of cumulative computational requirement of PMIS implemented with a bootstrap size of 0, 100, and 1 000.

selection of k input variables from an initial set of d candidates, the overall computational requirement of PMIS can be expressed in terms of the number of PMI evaluations as

$$O\{(kd - \frac{k(k+1)}{2} + Bk)n^2\}. \tag{6.18}$$

Consider, for example, the application of PMIS to the Friedman model datasets (5 and 6), which involved the selection of 5 of 10 candidate variables. The cumulative number of PMI evaluations over the six iterations of the PMIS algorithm are shown graphically in Figure 6.8, in terms of PMI evaluations for bootstrap sizes of 0, 100, and 1 000. The algorithm required only 45 evaluations to complete the selection without bootstrapping. For $B = 100$, this increased to a total of 645 evaluations of which 600 (93%) were performed as part of the bootstrap. This proportion increased to 99% of evaluations when the bootstrap size was increased to 1 000. This example represents the potential reduction in computational effort that could be achieved by using any of criteria E–H, since the computational effort of implementing each of these termination criteria is negligible compared to PMI estimation. In contrast, previous studies that have examined the improvement in the efficiency of PMIS gained by modifying the kernel used in density estimation, only reported a computational saving of 12% (*Bowden et al.*, 2005).

6.5.3 Linear versus non-linear input variable selection

PCIS performed well in terms of the frequency of correct selections, when applied to datasets generated by the linear models. In fact, the results in Figure 6.5(d) and Figure 6.4(d) show that PCIS yielded a higher frequency of correct selections than PMIS for each of the bootstrap-based criteria (A, B, C and D). It was also observed for these cases that the termination criteria that performed best overall were based on the $I_b^{(99)}$, rather than the more conventionally adopted $I_b^{(95)}$. PCIS performed worse than PMIS when the termination criteria based on analytical critical values were used (criteria E and F), though the results are consistent with those obtained for PCIS using the bootstrap-based criteria. Accuracy of bootstrap estimation of PMI aside, the result is perhaps indicative of the superiority of PMI for discerning conditional dependence in noisy data, which was mentioned in Section 6.2.

Another issue with PCIS was the poor performance of the Hampel test criterion (H). For nearly all instances of datasets generated by time-series models, this criterion failed to identify the first input and selection terminated, which resulted in a high frequency of under-selection. This result was attributed to potential masking due to the spread of (absolute) correlation. The absolute value of Pearson correlation is bounded on [0,1], and this made outliers more difficult to detect, since the distribution is less skewed. In contrast, I is bounded on [0, inf] so that outliers differ far more significantly from the median. This is evident in the comparison between PMIS and PCIS for the time-series models, for which PMIS achieved a high frequency of correct specification, and PCIS under-selected. A simple solution to this problem is to use a logarithmic transformation to add skewness, and for Gaussian data this would be an equivalent to estimating, I, as the following relationship holds:

$$I = -\frac{1}{2}\log(1 - R^2). \tag{6.19}$$

However, this is not an important issue, since the Hampel termination criterion was devised specifically for PMIS (for which the test worked well), and other termination criteria, such as analytic expressions for critical values of R used in criteria E and F, are considered more appropriate in this case of PCIS.

The justification for using a non-linear IVS algorithm for ANN development was evident from the comparison of the performance of PCIS for linear and non-linear datasets. As expected, PCIS performed relatively poorly in comparison to PMIS when applied to the datasets generated by non-linear models, as indicated by the high frequency of under-specification in Figure 6.6(d). Although not shown, the results for individual models showed that PCIS attained up to 70% correct selections when applied to data generated by the TAR2 time-series function. This surprising result was explained by further examination of the TAR1 and TAR2 time-series data. The degree of non-linearity of these time-series was assessed based on the Kaboudan fuzzy classification system (*Kaboudan*, 1999). Based on this analysis, the TAR1 time-series data were classified as non-linear with high level noise (NL-HN), and the TAR2 time-series classified as weakly linear, with white-noise (WL-WN). Hence, despite being nominally included as a non-linear test case, the TAR2 time-series proved to be quasi-linear, and PCIS actually performed worse on truly non-linear datasets than Figure 6.6 would suggest.

Overall, the results clearly revealed the unsuitability of PCIS as an IVS algorithm for the development of ANN models. Its use resulted in a high frequency ($\sim$70%) of under-specification, regardless of the termination criterion, when applied to non-linear datasets, which was an indication that the linear correlation coefficient was unable to identify one or more of the salient variables within the dataset. This provides a clear justification for the use of IVS algorithms based on non-linear measures of dependence for statistical models that are intended to represent non-linear processes, such as those found in many environmental systems.

6.5.4 Effect of sample size

It was evident from the overall results across the range of sample sizes, that sample size can have a significant effect on the accuracy of both PMIS and PCIS. Results for the smallest sample of size 50 indicate that the frequency of under-specified models was, on average, approximately 50%. Under-specification was most likely due to the apparent irrelevance of a “true” input variable for which the I' or R' was under-estimated. The equally poor performance of both IVS algorithms suggested that both statistical measures (PMI and linear correlation) were not able to reliably detect the relationships within the datasets based on such a small, noisy sample. As the length of datasets increased to 100–500 samples, the performance of the PCIS and PMIS algorithms improved significantly, in terms of selection accuracies, with a higher frequency of correct specification.

Differences between the relative performance of each criterion were also observed for small sample size, and this was considered to indicate the potential sensitivity of the criteria to the signal-to-noise ratio of the data. The tendency for most criteria was to under-specify the model. The results for the bootstrap based criteria (A–D) were highly variable, which can be attributed to the high variability of both relevance estimates and confidence bounds. The results for the AIC and Hampel criteria were also variable, and this can be similarly attributed to small-sample variance affecting the relative MI scores for candidates for individual experiments. Criteria E and F showed the least variability for small sample sizes, and consistently under-selected, which indicates clearly that the MI of relevant variables was under-estimated and failed to exceed the confidence bounds.

In conclusion, the application of filter-style IVS algorithms to smaller datasets should therefore consider the noise level in the data, since this will affect how well the statistical measures perform. However, this should not be regarded specifically as a weakness of the filter approach. The ability to determine relationships within data using MI is an estimate of the ability to construct a model to represent the relationship. Poor performance of PMIS may indicate that the development of an ANN model may also be problematic, and such considerations would need to be made during model development, because the degree of noise will also affect the ability of regression techniques to model the existing relationships.

6.6 Conclusions

The motivation behind this research was to formulate a more efficient means of correctly selecting input variables for artificial neural network (ANN) models of environmental processes. This has been achieved by the formulation of alternative termination criteria for an existing input variable selection (IVS) algorithm based on estimation of partial mutual information. Based on the performance of partial mutual information-based selection (PMIS) for the synthetic examples in this study, it was concluded that use of any of the three novel termination criteria provided a more accurate selection procedure than the traditional bootstrap approach. Furthermore, use of the novel criteria also substantially reduced the computational effort required to implement the IVS technique, to yield a more efficient selection procedure.

The relative merits and shortcomings of the different termination criteria introduced in this paper have a strong influence on the recommendation of the use of one of the novel criteria in favour of the others for a given situation. Given sufficient confidence that the data are known to be Gaussian, then the criteria based on the estimated critical values of mutual information would provide accurate selections, with the simplest implementation.

However, in real-world applications, the distributions may be unknown and the assumption of Gaussian data may not hold, so for this reason the AIC or Hampel criteria are considered to be more widely applicable. The AIC method provides a general measure of the trade-off between information gain and the complexity introduced to the modelling domain by the addition of input variables. This criterion lends itself to clear and simple interpretation, and is expected to provide consistent and reliable selection for any dataset. The method based on the Hampel outlier test was found to be sensitive to the relative number of salient variables among the candidates. Seeding the candidate set with noise variables overcomes this issue, although this is potentially creating additional computational burden rather than reducing it. In any case, provided masking does not interfere with the detection of outliers, as would be expected in most real-world time-series applications where the number of redundant variables is large, the Hampel outlier test-based termination criterion is expected to perform equally as well as the AIC-based termination criterion.

This study has also added further justification to the application of non-linear IVS techniques during the development of ANN models. The comparison between IVS selection techniques based on linear and non-linear measures of dependence clearly showed that linear measures failed to identify the complete set of input variables when the relationships are non-linear. Consequently, the application of linear IVS algorithms to the development of ANN models may potentially result in the omission of variables that are important for describing the behaviour of a given environmental process, and thus produce sub-optimal model performance.

Acknowledgements

This research was aided by the use of facilities provided by the South Australian Partnership for Advanced Computing (http://sapac.edu.au), and was partially funded by United Water International Pty Ltd and the Cooperative Research Centre for Water Quality and Treatment. The authors would also like to thank the anonymous reviewers for their comments, which have substantially improved the quality of this paper.

Chapter 7

Application of Partial Mutual Information Variable Selection to ANN Forecasting of Water Quality in Water Distribution Systems

Publication 3

Publication Details

This work has been published within the journal *Environmental Modelling and Software* as the following article:

> May, R. J., H. R. Maier, G. C. Dandy, and J. B. Nixon, Application of Partial Mutual Information Variable Selection to ANN Forecasting of Water Quality in Water Distribution Systems, *Environmental Modelling and Software*, 23, 1289–1299, 2008.

Although the manuscript has been reformatted in accordance University guidelines, and sections have been renumbered for inclusion within this thesis, the paper is otherwise presented herein as published.

Statement of Authorship

May, R. J. (Candidate)
Development and implementation of algorithms, experimental design, interpretation and analysis of results, manuscript preparation and corresponding author.

Signed: .... Date: ....

Maier, H. R.
Research supervision and review of manuscript.

Signed: .... Date: ....

Dandy, G. C.
Research supervision and review of manuscript.

Signed: .... Date: ....

Nixon, J. B.
Assistance with case study development and review of manuscript.

Signed: .... Date: ....

Abstract

Recent trends in the management of water supply have increased the need for modelling techniques that can provide reliable, efficient, and accurate representation of the complex, non-linear dynamics of water quality within water distribution systems. Statistical models based on artificial neural networks (ANNs) have been found to be highly suited to this application, and offer distinct advantages over more conventional modelling techniques. However, many practitioners utilise somewhat heuristic or ad hoc *methods for input variable selection (IVS) during ANN development.*

This paper describes the application of a newly proposed non-linear IVS algorithm to the development of ANN models to forecast water quality within two water distribution systems. The intention is to reduce the need for arbitrary judgement and extensive trial-and-error during model development. The algorithm utilises the concept of partial mutual information (PMI) to select inputs based on the analysis of relationship strength between inputs and outputs, and between redundant inputs. In comparison with an existing approach, the ANN models developed using the IVS algorithm are found to provide optimal prediction with significantly greater parsimony. Furthermore, the results obtained from the IVS procedure are useful for developing additional insight into the important relationships that exist between water distribution system variables.

7.1 Introduction

Maintenance of a residual chlorine concentration within a water distribution system (WDS) is a widespread strategy for ensuring the delivery of clean, safe drinking water to consumers. A chlorine residual provides a secondary barrier against harmful microbial pathogens, which may otherwise persist downstream of a water treatment plant. Furthermore, secondary disinfection inhibits the growth of biofilms and other microorganisms that degrade the quality of the water. However, recent trends in water quality management have resulted in stricter operational guidelines, with respect to allowable disinfectant concentrations; and consumer expectations have increased, with respect to taste and odour. Consequently, there has been an increased need for more effective management of WDS disinfectant residuals. The development of mathematical tools for describing the dynamics of WDS disinfectant residuals will undoubtedly be useful in this regard.

Forecasting the dynamics of disinfectant residual within a WDS can be a difficult task. Systems are typically large-scale with rapidly changing hydraulic conditions, which leads to significant spatial and temporal variation in detention time. The complex and non-linear behaviour of disinfectant residuals is also the result of multiple, interacting processes (*Serodes et al.*, 2001). Conventionally, disinfectant residual modelling has been based on a deterministic, simulation-based approach, in which a hydraulic model is coupled with a mathematical model of the intrinsic water quality processes. Water quality simulation models are useful for undertaking scenario (what-if) analysis and for design applications (*Walski et al.*, 2003). However, their successful application to operational management of water quality is often limited by several factors, including:

- *Hydraulic model.* The development of an accurate hydraulic model requires a combination of expertise and detailed information regarding the topology and hydraulic behaviour of a WDS (*Walski et al.*, 2003). In the absence of data, simplifying assumptions are often made regarding key hydraulic parameters (e.g. consumer demand patterns and mixing regimes within storage reservoirs).
- *Decay model.* Simple expressions, such as the first-order rate equation, are typical of models that have been developed to describe the kinetics of disinfectant decay within bulk water (*Kastl et al.*, 1999). Although increasingly sophisticated models have also been developed, these models still do not adequately represent water quality behaviour observed within water distribution systems. This may be due to the uncharacterised effects of biofilms, corrosion or other unobserved processes that occur within the WDS (*Clark and Haught*, 2005).

- *Computational constraints.* The computational effort required to undertake an extended-period simulation, especially when water quality is considered, can lead to long simulation times for even relatively small hydraulic systems. Many applications, such as real-time management, require model output within a much shorter time-frame than the simulation time. Furthermore, much of the computational effort expended is often wasted, since the simulation model fully describes the WDS, but it is only the input-output relationships at critical control locations that are required (*Polycarpou et al.*, 2002).
- *Demand forecasting model.* Forecasting water quality can become problematic using the simulation-based approach, since future demands need either to be known, or predicted with some certainty.

Polycarpou et al. (2002) conclude that *"What is needed is a simple analogy to the input-output simulation model that reflects the important physical and chemical processes, but without requiring a complex algorithm and* a priori *knowledge of the network hydraulics."* Hence, statistical modelling methods have attracted interest as a more expedient alternative to traditional, simulation-based techniques. Statistical modelling methods applied to disinfectant residual forecasting have been found to be highly suitable for this application, as well as for other applications in the field of water treatment and supply (*Rodriguez and Serodes*, 1999; *Baxter et al.*, 1999, 2001; *Milot et al.*, 2002; *Maier et al.*, 2004; *Gibbs et al.*, 2006; *Bowden et al.*, 2006). The relative abundance of hydraulic and water quality data generated by routine WDS monitoring, in contrast to the incomplete understanding regarding the processes that govern disinfectant decay, provides a great deal of justification for adopting the statistical modelling approach in preference to more traditional deterministic methods for the development of input-output water quality models.

Models based on various time-series forecasting techniques have been developed to forecast disinfection residuals, including: auto-regressive moving average (ARMA) (*Polycarpou et al.*, 2002), auto-regressive with exogenous inputs (ARX) and multiple linear regression (MLR) (*Rodriguez and Serodes*, 1999; *Gibbs et al.*, 2006; *Bowden et al.*, 2006), logistic regression (LR) (*Milot et al.*, 2002) and artificial neural network (ANN) models (*Rodriguez and Serodes*, 1999; *Serodes et al.*, 2001; *Gibbs et al.*, 2006; *Bowden et al.*, 2006). In particular, ANN models have proven to represent non-linear water quality dynamics more accurately and efficiently than their linear counterparts in a number comparison studies (see *Rodriguez and Serodes* (1999), *Bowden* (2003), or *Gibbs et al.* (2006) for examples).

This paper is concerned with the methodology that is applied to the development of ANN models for WDS disinfectant residual forecasting. In particular, the paper addresses the important issue of input variable selection (IVS). The importance of this task corresponds to the considerable negative impact on ANN performance that can result from the inclusion of variables that are either *irrelevant*, or *redundant*. However, although there is a well-justified case for careful consideration of the input variables that are chosen for ANN development, there is currently no consensus on how this task should be undertaken. An IVS algorithm based on the estimation of partial mutual information (PMI), which was introduced by *Sharma* (2000) and further developed by *Bowden et al.* (2005) and *May et al.* (2008a), is presented here as an improved methodology for performing the IVS task and is shown to lead to an improved overall framework for the development of ANN chlorine residual forecasting models.

The remainder of this paper is structured as follows. Section 7.2 describes the IVS problem and reviews current methods for the development of ANN models within the context of WDS disinfectant residual forecasting. A method for ANN development that includes the proposed IVS methodology is then described in Section 7.3. Sections 7.4 and 7.5 provide illustrative examples of the application of the methodology to the development of ANN models for forecasting chlorine residuals. Discussion of the results obtained is provided in Section 7.6. Finally, concluding remarks are given in Section 7.7.

7.2 Background

In statistical modelling, non-linear dynamic processes are approximated by a regression model of the general form

$$y(t+k) = F(y(t), \ldots, y(t-p), \mathbf{x}(t), \ldots, \mathbf{x}(t-q)). \tag{7.1}$$

Here, the model output y is predicted at some time, $t+k$, where $k > 0$. The model input comprises past observations (or, *lags*) of y and $\mathbf{x}$, which represents a multivariate set of exogenous input variables. The parameters p and q denote the model order (i.e. number of lags) with respect to the endogenous and exogenous variables. The functional form of F is initially unknown, and the goal of model development is to identify a suitable form based on a set of representative data.

Input variable selection is an important part of the identification of ANN models, since the form of the model is derived purely from the available data. In real-world applications, such as WDS analysis, there are potentially many variables that could be used as inputs to the ANN model. For example, a representative set of data for a typical WDS may contain observations of water quality parameters such as pH, turbidity, temperature, applied chlorine dose, and residual chlorine concentration; and hydraulic parameters such as flowrate, pump and valve status, and tank levels at points throughout the system. Considering the development of a dynamic model (i.e. including lags), the number of potential input variables can be quite large. However, for the development of ANN models, the minimum number of variables should be used as inputs to the ANN in order to: (i) increase computational efficiency; (ii) minimise redundancy; (iii) reduce noise; and (iv) increase the interpretability of the model (*Sindelar and Babuska*, 2004; *Back and Trappenberg*, 2001).

Past applications of ANNs to chlorine residual forecasting have utilised a variety of methods for undertaking IVS (*Rodriguez and Serodes*, 1999; *Serodes et al.*, 2001; *Gibbs et al.*, 2006; *Bowden et al.*, 2006). The Chlorcast© methodology of *Serodes et al.* (2001) is arguably the most comprehensive approach, and is similar to methodologies described in other closely related water supply applications such as coagulation process modelling (*Baxter et al.*, 2001). The Chlorcast© approach is shown conceptually in Figure 7.1(a), where the input variables comprise a window of endogenous and exogenous lags up to the optimal model order, d. The methodology implements a *wrapper* approach to model specification, in which the most appropriate model order, forecasting horizon and data interval are systematically determined based on trial-and-error analysis of trained ANN models. Given the many combinations of parameters, the implementation of Chlorcast© requires the training and evaluation of a potentially large number of ANN models. Although the use of the general regression neural network (GRNN) used in *Serodes et al.* (2001) reduces this requirement somewhat, the computational effort required for the more conventional multi-layer perceptron (MLP) would be significant due to both increased training times, and the requirement for additional experimentation to optimise the model architecture.

Furthermore, *Serodes et al.* (2001) observed *"that the larger the learning window the better the results..."* since an increased model order *"...conveys system dynamics in more detail"*. However, increasing the model order quickly increases the size of the input variable set, since lags of all parameters are added. For the example shown in Figure 7.1(a), an increase in model order from one to five increases the number of potential inputs from 5 to 25. Although the optimal model order is found (with respect to model prediction error), the larger window is more likely to contain many irrelevant or redundant input variables. The holistic analysis of the input variable set fails to consider that the optimum order with respect to individual parameters may differ, or that successive lags may be highly correlated. A lack of interpretability also results from the inclusion of irrelevant and redun-

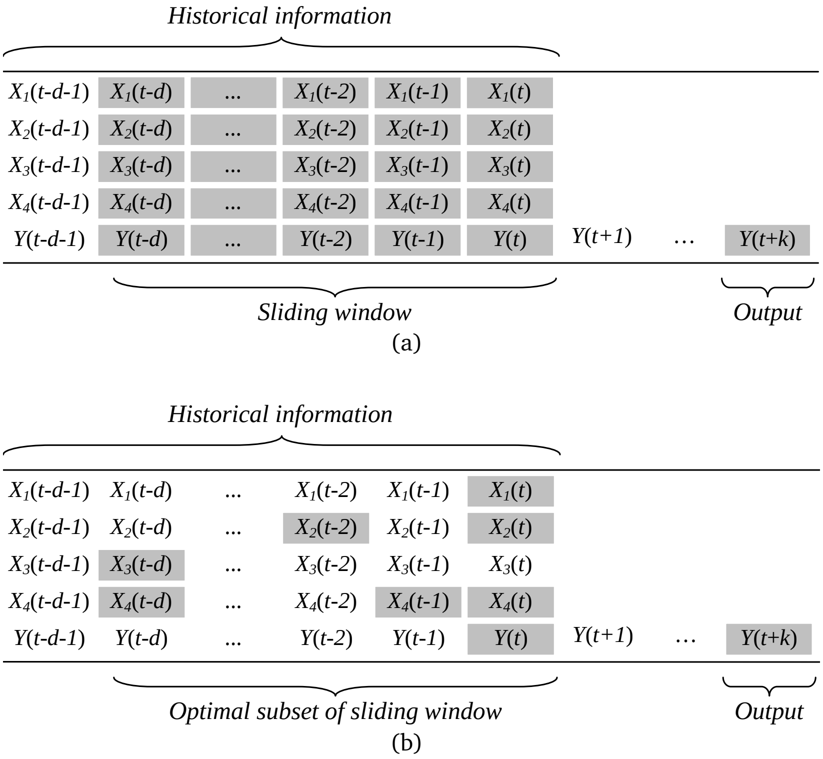

Historical information
$X_1(t-d-1)$ $X_1(t-d)$ ... $X_1(t-2)$ $X_1(t-1)$ $X_1(t)$
$X_2(t-d-1)$ $X_2(t-d)$ ... $X_2(t-2)$ $X_2(t-1)$ $X_2(t)$
$X_3(t-d-1)$ $X_3(t-d)$ ... $X_3(t-2)$ $X_3(t-1)$ $X_3(t)$
$X_4(t-d-1)$ $X_4(t-d)$ ... $X_4(t-2)$ $X_4(t-1)$ $X_4(t)$
$Y(t-d-1)$ $Y(t-d)$ ... $Y(t-2)$ $Y(t-1)$ $Y(t)$ $Y(t+1)$ … $Y(t+k)$
Sliding window
Output
(a)
Historical information
$X_1(t-d-1)$ $X_1(t-d)$ ... $X_1(t-2)$ $X_1(t-1)$ $X_1(t)$
$X_2(t-d-1)$ $X_2(t-d)$ ... $X_2(t-2)$ $X_2(t-1)$ $X_2(t)$
$X_3(t-d-1)$ $X_3(t-d)$ ... $X_3(t-2)$ $X_3(t-1)$ $X_3(t)$
$X_4(t-d-1)$ $X_4(t-d)$ ... $X_4(t-2)$ $X_4(t-1)$ $X_4(t)$
$Y(t-d-1)$ $Y(t-d)$ ... $Y(t-2)$ $Y(t-1)$ $Y(t)$ $Y(t+1)$ … $Y(t+k)$
Optimal subset of sliding window
Output
(b)

Figure 7.1: Conceptual approach to statistical regression, with inputs comprising (a) a complete window of endogenous and exogenous lags (after *Serodes et al.* (2001)), and (b) an optimal subset of selected variables.

dant input variables, which is often a criticism of the ANN modelling approach in general (*Baxter et al.*, 2001; *Serodes et al.*, 2001).

Finally, methodologies like Chlorcast© do not provide guidelines as to which WDS parameters are relevant for a given forecasting application. Rather, this decision is usually based on experience and judgement (*Baxter et al.*, 1999; *Serodes et al.*, 2001). Applying the methodology is therefore difficult when there is insufficient *a priori* information available, or the system under consideration is too complex for the modeller to grasp intuitively. Necessary parameters could easily be excluded, or superfluous parameters included mistakenly, by an inexperienced modeller—either of which would reduce model performance. However, the aim of establishing ANN development frameworks is to reduce the need for judgement as much as possible, and instead rely more upon analytical approaches (*Maier*, 2006).

In conclusion, it is proposed that the overall development of ANN water quality models can be improved by applying a suitable algorithm for IVS as the first step in model development. The application of an analytical approach, such as the PMI-based algorithm presented previously by *May et al.* (2008a), would enable the selection of only input variables that are significant (i.e. useful for modelling the output) and the exclusion of variables that are redundant. The amount of trial-and-error during model development would be significantly reduced, and the resulting parsimonious input variable set—which is shown conceptually in Figure 7.1(b)—would yield improved performance. Furthermore, the information gained regarding the significant input variables and their relative influence on the output variable can provide valuable insight into how predictions are generated by the ANN, and can be used to better direct ongoing monitoring and data collection efforts.

In this paper, the above benefits are demonstrated by examining two case study examples, where the PMI-based algorithm, hereafter referred to as partial mutual information-based selection (PMIS), is incorporated into the development of ANN models for forecasting residual chlorine concentrations. The utility of the algorithm is determined from the results of the case studies, and the overall ANN development is compared with the Chlorcast© ANN development methodology by considering factors such as (i) the impact of IVS on ANN performance, in terms of both prediction accuracy and parsimony; (ii) the relative ease of model development; and (iii) the interpretability of models.

7.3 Methodology

In this paper, the application of the IVS methodology to two example case studies is described to illustrate the utility of the PMIS algorithm in the development of ANN residual chlorine forecasting models. The first example is the development of ANNs for 1-hour ahead forecasting of chlorine within the simulated Cherry Hills–Brushy Plains WDS, which features complex dynamics due to multiple chlorine sources and intermittent raw water supply. In the second example, the IVS methodology is further validated on the task of 24-hour ahead ANN forecasting of chlorine within a real-world WDS in Myponga, South Australia. The following describes the methodology that was applied to develop ANN models for both case studies.

7.3.1 Model architecture

The ANN architecture adopted in this case study was the general regression neural network (GRNN), which is a class of ANN that was first introduced by *Specht* (1991) as a neural network *paradigm* for kernel regression. The GRNN is a probabilistic neural network (PNN), and is similar to radial basis function (RBF) networks. In fact, the GRNN can be considered to be a special case of a normalised RBF network, in which a basis function is centered on each training data input vector, and weighted by the corresponding training output (*Sarle*, 1997).

The architecture of the GRNN is shown in Figure 7.2. The input x is fully connected to each pattern layer node j, for which an activation $a_j(x)$ is determined based on a kernel function centered on a training input vector z_j. The Gaussian kernel function, in which the Euclidean distance metric determines the activation, is typically used as the activation function in the pattern layer. In this case the activation is given as

$$a_j(x) = \exp \frac{-\|x - z_j\|^2}{2h^2} \tag{7.2}$$

where h[1] is the GRNN bandwidth, or *smoothing parameter*. The activation of each pattern layer node is passed to the two nodes in the summation layer, which each generate weighted sums of the pattern node activations. The connection weights between the num summation node and the pattern layer are the values y_j that correspond to each z_j, so that the activation of the num summation node is given as

$$num = \sum_{j=1}^{n} y_j a_j \tag{7.3}$$

The connection weights between the pattern layer and the den summation node are equal to 1, and the activation at this node is given as

$$den = \sum_{j=1}^{n} a_j \tag{7.4}$$

In the output layer, the ratio of the activations of the num and den nodes determines the network output, so that the global transfer function $G(x)$ that is

[1] Often the GRNN bandwidth is referred to as σ, however the notation adopted here is kept consistent with kernel regression literature, and to avoid confusion with the standard deviation.

achieved by the GRNN can be written as,

$$G(x) = \frac{\sum_{j=1}^{n} y_j a_j}{\sum_{j=1}^{n} a_j}, \tag{7.5}$$

which is simply the kernel estimate for $E(y|x)$, that is, the conditional expectation of y given x.

In comparison to the more conventional multi-layer perceptron (MLP), the GRNN has both advantages and disadvantages. The GRNN uses memory based (or, *lazy*) learning, and therefore has an increased memory requirement to store the training data and a greater computational requirement when querying the network than a MLP, which more efficiently stores the relationship learnt during training within the architecture. However, the GRNN is much faster to develop, as it has only a single parameter—the kernel bandwidth—that needs to be learnt during training; and the network architecture is fixed, which avoids the need to train multiple models to optimise the network architecture (*Specht*, 1991; *Sarle*, 1997). The GRNN is best suited to applications where the distribution of the data is smooth and continuous. Hence, the GRNN has previously been found to be a suitable ANN architecture for the modelling of water quality within water distribution systems (*Serodes et al.*, 2001; *Bowden et al.*, 2006). Given its suitability for the type of data, and its simple and efficient development, the GRNN was considered a good choice for this application. Finally, the use of the GRNN provides a direct basis for comparison with the Chlorcast© methodology, which also utilised this network architecture.

7.3.2 Input variable selection

PMIS was applied to determine an optimal subset of the candidates to use as inputs to the GRNN model. The development of ANN models implicitly assumes a degree of non-linearity within the relationships between variables, and PMIS is therefore highly suited to this application, as it uses mutual information (MI) to measure the relevance of candidates. As has been demonstrated previously, more conventional methods that are based on linear correlation may fail to identify important input variables due to their inability to identify non-linear relationships, and are therefore considered unsuitable for ANN development (*May et al.*, 2008a).

In PMIS, the candidate set initially contains all lags of hydraulic and water quality parameters that may potentially be included as inputs to the ANN model. The selected input variable set is initialised to the null set (i.e. no input variables are initially selected). The PMIS algorithm then proceeds by iteratively selecting inputs that have the maximum relationship strength with the output variable, conditional on any previously selected inputs. Full details of the partial mutual information-based selection (PMIS) algorithm are presented in *May et al.* (2008a). However, the procedure can be briefly summarized as follows:

```
i. Initialise candidate and selected input variable sets.
ii. Filter observations of the output by subtracting relationship with currently
    selected inputs.
iii. For each candidate input
    a) Filter candidate by subtracting relationship with currently selected inputs.
    b) Estimate the PMI between the filtered output and candidate.
iv. Find the candidate that maximises PMI.
v. Determine significance of the candidate-output PMI.
vi. If the candidate is significant then
    a) Move candidate to input variable set.
    b) Return to Step ii.
    Else terminate selection.
```

The PMIS termination criterion, which determines whether a given candidate is significant, can influence the number of selected variables. Two recently developed criteria, based on the Akaike Information Criterion (AIC) and Hampel test (full details of which are given by *May et al.* (2008a)) were applied to determine when to terminate selection. Benchmarking on synthetic problems has found that these two criteria are both efficient and do not require assumptions regarding the underlying distribution of the data. Hence they were adopted for this study to further validate their suitability for this application.

The PMIS approach offers a simpler, and yet more flexible model development framework, in comparison to existing methods, such as Chlorcast©. Firstly, the algorithm effectively removes any uncertainty regarding the relevance of available WDS parameters that might be included as ANN input variables. Such parameters can be initially included as candidates, since irrelevant parameters will ultimately be identified and ignored by the selection algorithm.

Second, in terms of selecting lag variables, the modeller only needs to specify the maximum order of lag to be included in the candidate set. Provided that sufficient lags are initially included, the order of the system will be identified through the selection of relevant lag variables. This not only results in a more parsimonious model overall, but also eliminates the need to determine the optimal size of the lagging window by trial-and-error. Furthermore, the proposed IVS approach can allow for some initial conservatism in the face of uncertainty regarding the order of the system under consideration, without sacrificing model performance. Conservatism in the conventional ANN approach would dictate that variables be considered if there is a chance that they might yield some required information regarding the output variable, as the ANN will either use or disregard variables as a result of the learning. This detracts from the performance of the model due to a larger input space and a more complex network architecture. If the proposed IVS algorithm is used, performance of the model is not compromised by conservatism, since although many irrelevant candidates can be initially included without prior knowledge of their informativeness, they will ultimately be rejected by the IVS algorithm, leaving the optimal input variable set.

Finally, the optimal set of available input variables is guaranteed to be selected for any forecast disinfectant residual, regardless of the location and forecasting horizon. Although the accuracy of ANN models is expected to differ as the forecast horizon changes, the IVS approach allows the development of the best ANN model for the application at hand. Such added flexibility is an attractive benefit of adopting this approach, since various modelling applications may demand different forecasting horizons.

In order to demonstrate the benefits of applying PMIS, input variables sets were also defined for models based on the range of inputs that would be selected using the Chlorcast© methodology. The minimal number of input variables corresponds to all parameters at time t, that is, using no lagged variables. The maximum number of inputs possible corresponds to all lagged variables within a pre-defined window. The hypothesis is that the inclusion of more input variables improves model performance, although sufficient performance is quite possible using the minimalist input variable set. In this study, these extreme sets are used as a basis for comparing the selections that are generated by the PMIS algorithm.

7.3.3 Data sampling

Like other ANN architectures, the GRNN network is susceptible to over-fitting, since a perfect fit to the training data can be obtained with a sufficiently small bandwidth. Hence, hold-out cross-validation was used during training, in which the optimal bandwidth was determined by minimisation of the prediction error for the testing data, to ensure that the best degree of generalisation was achieved. However, since the test data are used to determine the optimal bandwidth for the GRNN, the model is potentially optimistically biased towards the test results, and another hold-out is required to perform final validation of the model performance, to confirm the model is able to generalise. Consequently, the development of ANN models required that the available modelling data be sampled into three smaller subsets for training, testing, and validating the network. In this study, the respective proportions of samples allocated to each of these subsets were:

- 64% training,
- 16% testing, and
- 20% validation.

In order to eliminate any potential variance or bias in model performance that could be attributed to the sampling procedure, ensemble training was used, in which an ensemble of GRNN models was trained based on independent resamplings of the data. Uniform random sampling of the data was used to sample 100 instances of data subsets according to the specified proportions. A GRNN was then trained on each instance of training and test data, and queried against the corresponding validation set. The aggregate (mean) validation performance for all models then provided an indication of the expected model performance, and the variance could confirm the confidence bounds to allow comparisons between different models. Ensemble training is an effective means of minimising sample bias and variance in the performance of models, which can potentially be introduced by the hold-out cross-validation procedure (*Anctil and Lauzon*, 2004). All data are used during training, including extreme cases, so that no information is lost due to the hold-out.

7.3.4 GRNN training

The training of the GRNN essentially represents a one-dimensional optimisation problem, in which the network error, E, is minimised with respect to the bandwidth, h, where E is the test error since, in this case, hold-out cross-validation is used. This one-dimensional optimisation of the bandwidth was performed using an implementation of Brent's algorithm (*Press et al.*, 1992), which arrives at an optimal bandwidth faster than techniques such as gradient descent or hill-climbing. Similar to interval halving and other bracketing optimisation algorithms, Brent's algorithm shrinks the interval between a bracketing pair of points (points that lie either side of the optimum bandwidth) by iteratively guessing the optimum based on a parabolic fit to the error function evaluated at these points. In this regard, Brent's algorithm is able to exploit the smoothness of the GRNN error function, $E(h)$ near the optimal value of h (*Bowden et al.*, 2006). The advantage of Brent's algorithm, in comparison to conventional techniques, is that the GRNN can be trained within a very few iterations. However, an initial bracketing of the optimal bandwidth is required to initialise the algorithm.

Previously, *Bowden et al.* (2006) utilised a trial-and-error based on Golden search to determine this initial bracketing. In this study, the interval $[\varepsilon, \varphi h^*]$ was used, where ε is a small value near, but greater than, zero; φ is the Golden ratio[2] (~ 1.618); and h^* is the Gaussian reference bandwidth (*Scott*, 1992). Since h^* is often close to, and typically greater than the optimal value, this was found to be a suitable means of intialising Brent's algorithm that avoided the need for a trial-and-error approach.

7.3.5 Performance criteria

A multi-criteria approach was adopted for assessing the models developed, in which model performance was evaluated using several statistical error functions and goodness-of-fit measures, including the root mean squared error (RMSE), the mean absolute error (MAE), the mean relative error (MRE), the coefficient of determination (r^2), and the Akaike information criterion (AIC) (*Akaike*, 1974):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{7.6}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{7.7}$$

[2] The optimisation initialisation is essentially the first iteration of the Golden search algorithm (see *Press et al.* (1992)) with h^* as the initial trial solution.

$$\mathrm{MRE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \tag{7.8}$$

$$r^2 = \frac{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)\left(\hat{y}_i - \tilde{y}\right)}{\sqrt{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2 \sum_{i=1}^{n}\left(\hat{y}_i - \tilde{y}\right)^2}}, \tag{7.9}$$

$$\mathrm{AIC} = n\log\left(\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2\right) + 2p. \tag{7.10}$$

Here y_i and $\bar{y}$ are the observed and mean values of the actual chlorine time-series, resepectively; $\hat{y}_i$ and $\tilde{y}$ are the corresponding observed and mean values of the predicted chlorine time-series, respectively; n is the total number of observations; and p is the number of model parameters.

The first four are typical performance criteria used in existing frameworks for statistical model development to evaluate the forecasting accuracy of the models (*Serodes et al.*, 2001; *Baxter et al.*, 2000). In this study, the RMSE was the primary measure of forecasting error, as it was also used as the training error. The MAE provided a secondary indication of the expected magnitude of the error in terms of the units of the output. The MRE is also calculated, as it provides a more moderate indication of the error, due to its reduced sensitivity (in comparison to the RMSE) to errors at either extreme of the output variable range (*Karunanithi et al.*, 1994). The r^2 provides an indication of the similarity between actual chlorine residuals and model forecasts (*Serodes et al.*, 2001).

The AIC was adopted as an additional performance criterion, given the focus of this paper on comparisons between the composition of the input set and model accuracy. The AIC is a function of the RMSE, however it penalises the selection of superfluous input variables that do not significantly improve model performance. The set of input variables corresponding to the minimum AIC represents the optimal trade-off between the size of the input variable set and forecasting accuracy. Reducing the size of the input set is particularly important for the GRNN architecture, since kernel regression estimation rapidly becomes less accurate for a finite training data sample as the number of dimensions (i.e. inputs) increases beyond the range of six to ten variables (*Scott*, 1992).

7.4 Cherry Hills–Brushy Plains WDS Example

The following example describes the application of the methodology described in Section 7.3 to a simulated water distribution network. In this meta-modelling approach, the simulation model represents an *unknown process*, which is observed during operation to generate pseudo-historical data, that can then be used to develop a statistical model to represent the process. The simulated model is realistic enough to generate data with the inherent complexity of data for the actual water distribution system. However, "historical" data can be readily generated in order to test and demonstrate the GRNN development methodology. In addition, it enables variables to be varied over their full range, which is generally not feasible for a real WDS. Such an approach has been used previously for ANN development, and can allow testing of GRNN deployment and development of control applications, which often need to be tested within a simulated environment prior to deployment within the physical system (*Broad et al.*, 2005; *Raduly et al.*, 2007).

7.4.1 System Description

The Cherry Hills–Brushy Plains WDS is shown in Figure 7.3. This network was selected as a test case as it has been used previously for the evaluation of other hydraulic and water quality optimisation applications (*Boccelli et al.*, 1998). The main features of the WDS are: the pumping station, which operates on a six-hour cycle (i.e. six hours on, six hours off) to supply water to the system at sufficient flowrates to meet the average daily demand; a common inlet-outlet storage tank, which provides a buffering capacity for demand when the pumping station is off; and six potential booster chlorination points (A–F), to maintain the minimum required residual chlorine concentration at the extremeties of the WDS.

In this example case study, GRNN models of the network were developed to provide a 1-hour forecast of chlorine concentration at Node 36, using a historical database generated by simulation of the network. This case study therefore represents the application of ANN models to a complex multiple-input/single-output (MISO) system, where the aim is to map the relationship between the chlorine concentration at a single downstream location and injection rates at multiple, upstream booster locations.

7.4.2 Synthetic data generation

In order to generate data for model development, a quasi-dynamic water quality simulation of the Cherry Hills–Brushy plains network was undertaken. During the simulation the rate of chlorine injection into the system was allowed to vary with time, and the input-output response data with respect to chlorine concentrations were observed to generate data for ANN development. Optimal chlorine injection schedules for each booster location within this system have been determined previously by *Boccelli et al.* (1998). Based on these schedules, the chlorine injection rates at each booster location were randomly adjusted every six hours to lie within $\pm 20\%$ of the optimal dose for the given period. Bulk chlorine decay within the network was modelled as a first-order decay function, with a decay coefficient of -0.5 days^{-1}; and wall decay was neglected.

Simulation of 30 days of operation was performed using the object-oriented toolkit for EPANET (OOTEN) (*Van Zyl et al.*, 2003). For the purpose of this application, a Query class was added to the existing OOTEN library to facilitate the generation of data in a format suitable for ANN model development. During the simulation, multiple Query objects were used to poll individual water quality and hydraulic parameters at a short, regular time-interval of five minutes; and report an aggregate value (in this case, the average) at a longer time-interval of one hour.

Although the simulated WDS could be fully observed, queries were restricted in this case study to a set of key hydraulic and water quality parameters in order to reflect monitoring practices that are typical of real-world water distribution systems. This set of parameters included chlorine concentrations immediately downstream of booster locations A, B, C and F (i.e. the applied dose at each location), pumping station flow, trunk main flow at the mid-point of the system (Pipe 12) , and tank level.

Due to the nature of the metamodelling approach, very little data processing was required. However, the first 48 hours of data were discarded to account for any effects caused by the initialisation of the simulation model. Lags of each parameter for up to 48 hours into the past were considered to sufficiently capture detention times within the system, and a corresponding lead value of chlorine at Node 36 was also generated to provide forecast targets. Consequently, the processed database available for model development comprised a total of 720 observations of 384 candidate input variables and one forecast target. A statistical summary of the data collected for each of the unique WDS parameters observed is given in Table 7.1, which formed the historical database used for the development of the

ANN forecasting model.

7.4.3 Selected input variables

The results of input variable selection using the PMIS algorithm are summarised in Table 7.2, which shows the input variable selected and the corresponding PMI at successive iterations of the selection procedure. The AIC-based criterion terminated selection after four selections, as indicated in Table 7.2, and the Hampel test resulted in the selection of a total of six input variables.

Models with inputs selected by PMIS are denoted as Models A and B, where the AIC or Hampel test termination criteria were used, respectively. Input variable sets comprising of all available parameters at time t only, and for all available lags $(t, \ldots, t-48)$ were considered for comparison purposes. The models corresponding to the these input variable sets are denoted as Models C and D, respectively. These sets of inputs represented the smallest and largest input variable sets that would be selected according to the Chlorcast© methodology—given the initial candidate set; and which corresponded to the minimum and maximum amount of available information regarding the dynamics of the WDS, respectively. The input sets corresponding to all models are summarised in Table 7.3.

7.4.4 Model performance

Table 7.4 summarises the performance of the GRNN models for the test data. The values correspond to the mean observed for all individual GRNN networks within the ensemble of 100 networks, which were each trained on independent instances of training, test and validation data. The corresponding variability of these results is indicated by the standard deviations, which are the values in parentheses.

Model B, for which inputs were selected using PMIS in conjunction with the Hampel test-based termination criterion, performed the best, in terms of accuracy, with the lowest average prediction error for all error measures. Model A, for which inputs were selected using PMIS with the AIC-based termination criterion, was the third best model in terms of accuracy, but this model also utilised the fewest inputs and represented the optimal trade-off between the size of the input set and model accuracy measured by the AIC. It should be noted that, in general terms, all models performed well (with the exception of Model C), although the historical data generated by the meta-modelling approach are free of noise and

Table 7.1: Summary of historical data generated by simulation of the Cherry Hills–Brushy Plains WDS.

Parameter	Variable	Min.	Max.	Ave.	St. dev.	25%-ile	75%-ile	Interquartile Range
Chlorine at Node 36 (mg/L)	C_{36}	0.17	1.79	0.59	0.54	0.22	1.25	1.03
Chlorine at Node A (mg/L)	C_{A}	0.22	5.99	1.43	1.36	0.34	1.75	1.40
Chlorine at Node B (mg/L)	C_{B}	0.28	8.06	1.52	1.50	0.38	1.76	1.38
Chlorine at Node C (mg/L)	C_{C}	0.08	0.91	0.23	0.20	0.11	0.29	0.18
Chlorine at Node F (mg/L)	C_{F}	0.03	0.34	0.08	0.07	0.04	0.10	0.06
Tank level (m)	H	59.87	68.06	64.31	2.65	61.89	66.65	4.76
Pump flowrate (L/min)	Q_1	0.0	694.4	310.3	316.1	0.0	666.6	666.6
Pipe 12 flowrate (L/min)	Q_2	-235.1	640.6	204.9	328.9	-121.7	561.1	682.8

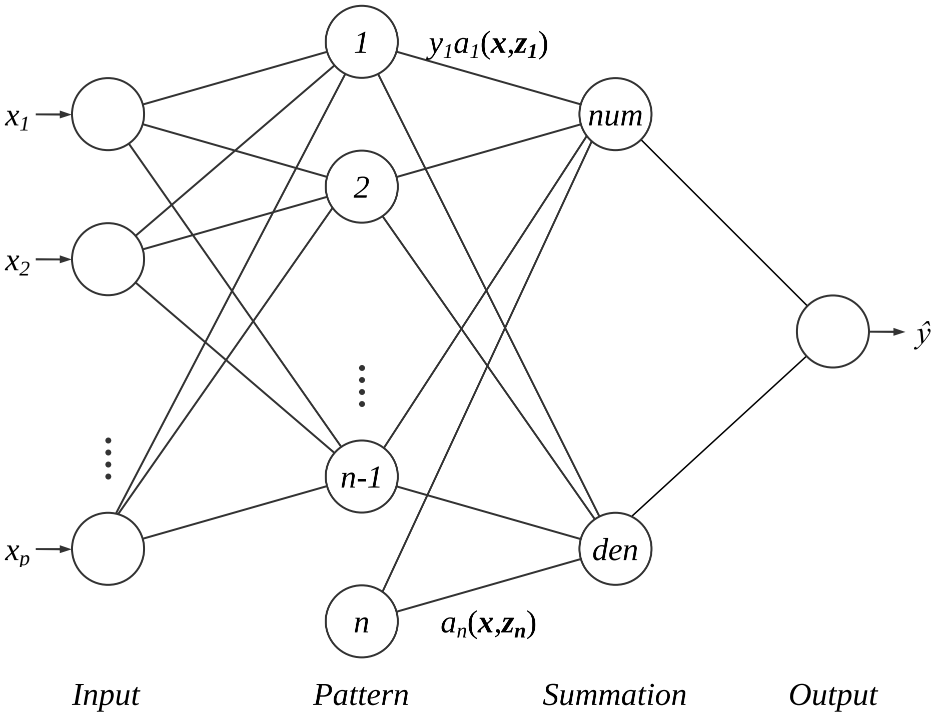

x_1
x_2
x_p
1
2
n-1
n
$y_1 a_1(\mathbf{x},\mathbf{z_1})$
$a_n(\mathbf{x},\mathbf{z_n})$
num
den
$\hat{y}$
Input
Pattern
Summation
Output

Figure 7.2: Architecture of the general regression neural network (GRNN).

Table 7.2: PMIS analysis of input variables for the Cherry Hills—Brushy Plains WDS case study.

Iteration	Candidate	PMI	Termination
1	$C_{36}(t-47)$	1.758	
2	$C_{36}(t-23)$	0.292	
3	$C_{\text{C}}(t-42)$	0.260	
4	$C_{\text{A}}(t-20)$	0.223	AIC
5	$C_{\text{A}}(t-43)$	0.214	
6	$C_{\text{C}}(t-33)$	0.198	Hampel test

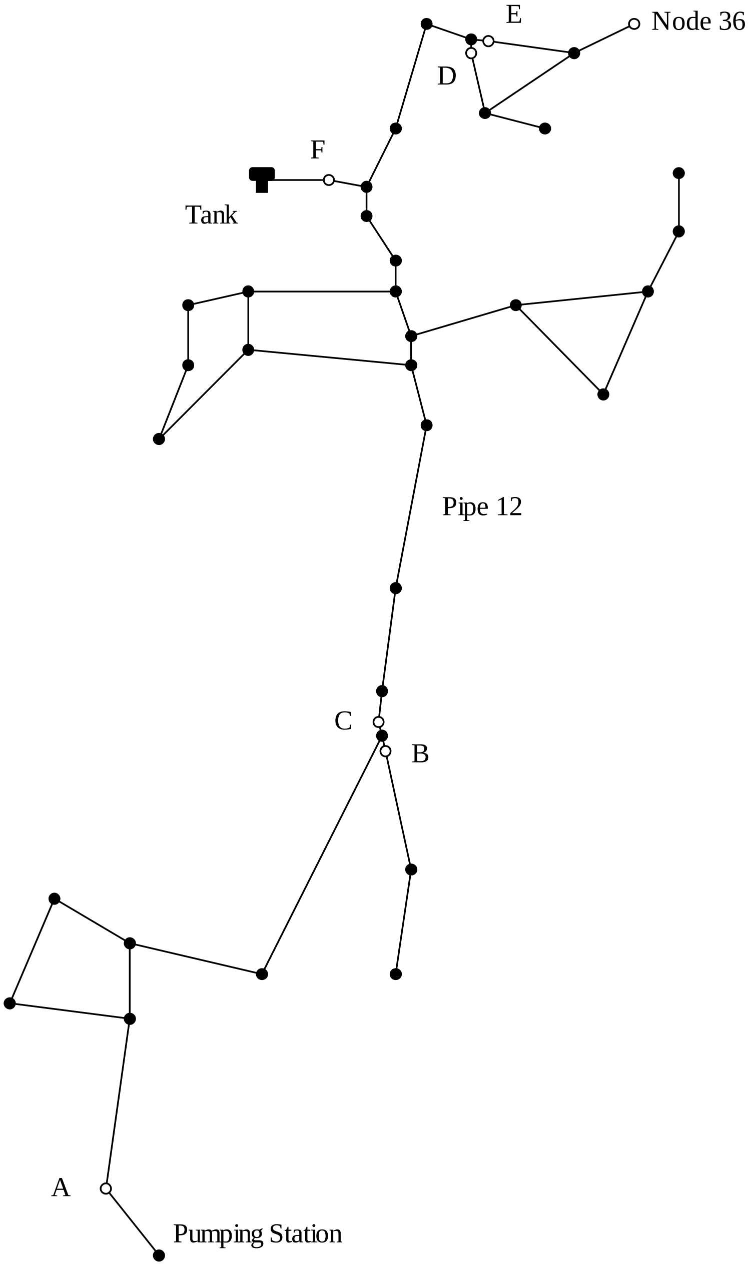


Figure 7.3: Topology of the Cherry Hills–Brushy Plains WDS.

Table 7.3: Summary of input variables for GRNN models of the Cherry Hills–Brushy Plains WDS

Model	# Inputs	Inputs
A	4	$C_{36}(t-23), C_{36}(t-47), C_{\mathrm{C}}(t-42), C_{\mathrm{A}}(t-20)$
B	6	$C_{36}(t-23), C_{36}(t-47), C_{\mathrm{C}}(t-42), C_{\mathrm{A}}(t-20), C_{\mathrm{A}}(t-43), C_{\mathrm{C}}(t-33)$
C	8	$C_{36}, C_{\mathrm{A}}, C_{\mathrm{B}}, C_{\mathrm{C}}, C_{\mathrm{F}}, H, Q_1, Q_2\ \forall(t)$
D	384	$C_{36}, C_{\mathrm{A}}, C_{\mathrm{B}}, C_{\mathrm{C}}, C_{\mathrm{F}}, H, Q_1, Q_2\ \forall(t,\ldots,t-48)$

Table 7.4: Test performance for 1-hour forecasts of residual chlorine within the Cherry Hills–Brushy Plains WDS.

Model	RMSE* (mg/L)	MAE* (mg/L)	MRE*	r^{2*}	AIC
Model A	0.0067 (0.0013)	0.0043 (0.0006)	0.0541 (0.0038)	0.9958 (0.0014)	5.7
Model B	0.0039 (0.0016)	0.0028 (0.0004)	0.0427 (0.0038)	0.9983 (0.0027)	9.2
Model C	0.0175 (0.0080)	0.0088 (0.0058)	0.0981 (0.0982)	0.9542 (0.0974)	14.5
Model D	0.0050 (0.0013)	0.0031 (0.0005)	0.0395 (0.0040)	0.9977 (0.0010)	765.4

*Values in parentheses denote standard deviation.

therefore results are expected to be of a high quality. In relative terms, there were significant differences in the performances of the four models. Similar ranges in model accuracy were reported in previous studies by *Serodes et al.* (2001) and *Bowden et al.* (2006) who reported r^2 values of 0.95–0.98 for test and validation data. A t-test based on the trials conducted in this study confirmed that the observed differences in the mean performance of each model were significant at a 5% (two-tailed) confidence level.

The need to include dynamic variables is evident from the relatively poor performance of Model C, for which the average RMSE (0.0175) was significantly greater than for all other models. In fact, a consistent trend was observed across all performance criteria, which clearly confirms the improved quality of forecasts generated by Model B when compared with Model C.

Model D, which utilised all available lags, was the second best performing model

Table 7.5: Validation performance for 1-hour forecasts of residual chlorine within the Cherry Hills–Brushy Plains WDS.

Model	RMSE* (mg/L)	MAE* (mg/L)	MRE*	r^{2*}	AIC
Model A	0.0092 (0.0083)	0.0048 (0.0018)	0.0561 (0.0067)	0.9854 (0.0496)	5.9
Model B	0.0060 (0.0059)	0.0032 (0.0009)	0.0450 (0.0052)	0.9932 (0.0188)	9.5
Model C	0.0197 (0.0089)	0.0095 (0.0059)	0.1043 (0.0918)	0.9516 (0.0747)	14.6
Model D	0.0056 (0.0025)	0.0033 (0.0006)	0.0418 (0.0051)	0.9968 (0.0047)	765.5

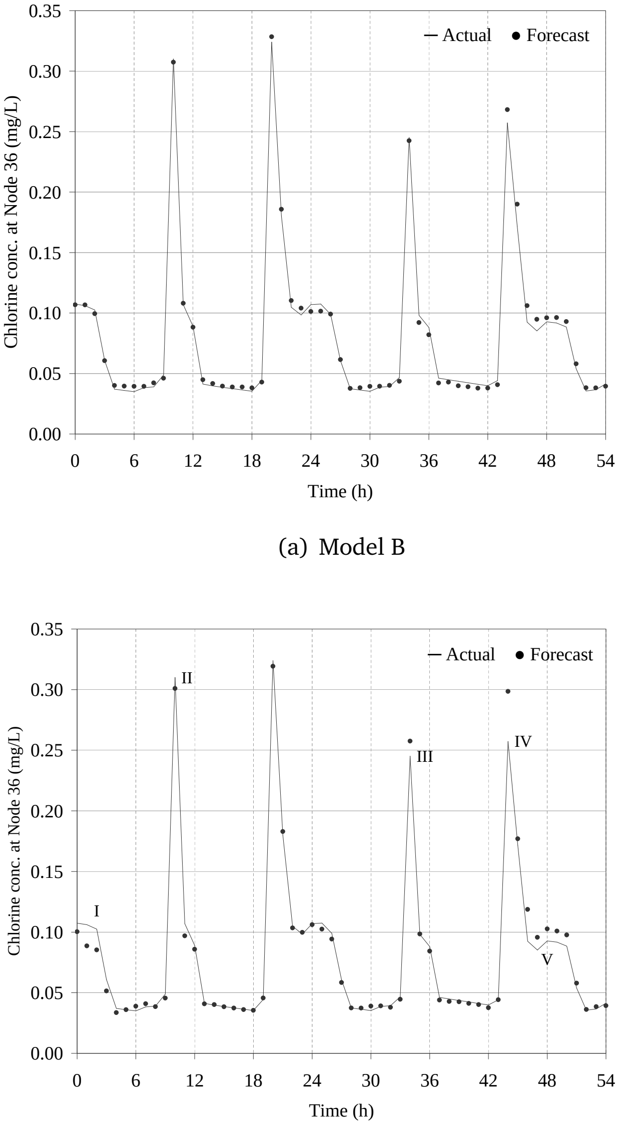
*Values in parentheses denote standard deviation.

in terms of prediction accuracy, with an RMSE (0.0050), whic is comparable to that of Model B. However, a comparison between Models B and D demonstrates the advantage of using the PMIS algorithm to select a subset from within the available sliding window to more efficiently represent the dynamic system. The high AIC obtained for Model D (~ 765) indicates the low efficiency of this particular GRNN due to the large number of input variables.

Results for the validation data are summarised in Table 7.5. It can be seen that the validation errors are similar to the test errors, indicating that the GRNN models developed have achieved good generalisation. It can be concluded that the assessment of the models based on the test results is therefore valid. A comparison of the validation time-series plots for Models B and C (Figure 7.4) provides a clearer indication of the difference in performance between these two models. The forecast time-series plot for Model C (Figure 7.4(b)) shows several regions (labelled I–V) of poor performance. In particular, it was observed that Model C had difficulty forecasting the sharp daily peaks (II, III, and IV) in chlorine concentration. In contrast, the trend for Model B (Figure 7.4(a)) shows that the inclusion of additional input variables, specifically selected using PMIS, resulted in improved forecasts in the labelled regions.

7.5 Myponga WDS Example

The following describes the application of the GRNN development methodology described in Section 7.3 to an actual water distribution system.



(a) Model B

(b) Model C

Figure 7.4: Actual and forecast validation time-series for chlorine at Node 36 of the Cherry Hills–Brushy Plains WDS for (a) Model B, for which input variables were selected using PMIS, and (b) Model C, for which no lags were selected as input variables.

7.5.1 System Description

The Myponga water treatment plant is managed and operated by United Water International Pty Ltd under contract with the regional regulatory authority, SA Water. The plant is situated 60 km to the south of Adelaide, South Australia, adjacent to the Myponga Reservoir, from which the plant is supplied with raw water. The treatment process combines alum floculation with dissolved air flotation and rapid dual-media filtration. Post-filtration, the pH is corrected by caustic dosing, and the filtered water is then disinfected by a chlorine injection system that is flow-paced to achieve a set-point free chlorine concentration, which is specified by the plant operator. Following a short detention time in a contact tank, the finished water flows into the filtered water storage tanks, from which the water flows under gravity via a trunk main, which supplies several branched reticulation systems. The plant does not have provision for booster chlorination at the outlet of the filtered water storage tanks. However, the primary chlorinator set-point, which is determined by the WTP operators, is considered to provide a sufficient dose to maintain minimum free chlorine residuals at the extremities of the distribution system. Due to this configuration, fluctuations in detention time within the filtered water storage tanks can have a significant impact on the free chlorine residuals that are observed downstream.

In this case study, ANN models were developed to forecast residual concentrations of free chlorine 24-hours in advance, at a monitoring location that was situated at a branch location on the trunk main, approximately 20 km downstream of the filtered water storage tanks.

7.5.2 Data collection and pre-processing

A six-month period of monitoring and data collection was undertaken between December 2002 (Summer) and July 2003 (Winter) to obtain a database for model development. Several sources of on-line operational data were available from routine monitoring, including: turbidity of the filtered water, corrected pH, free chlorine residual immediately downstream of the primary injection point (surrogate for applied primary dose), free chlorine downstream of the filtered water storage tanks, filtered water storage outlet flow, and filtered water storage level. An additional sensor was temporarily installed at the downstream forecasting location on the trunk main, which provided on-line measurement of both free chlorine and water temperature. A statistical summary of the data collected for each of the monitored hydraulic and water quality parameters is provided in Table 7.6. Although the data do not span the full year as recommended by *Serodes* *et al.* (2001), the period was considered sufficient to build a training database that captured both summer and winter seasonal operational conditions, that is, the range of the data collected encompassed all possible extreme operating conditions.

The raw data interval varied across different parameters from 10 to 15 minutes, although in this study the data were aggregated during pre-processing to the average over an hourly interval for all variables. The data were then examined to check for any erroneous values, or gaps, that may have been caused by instrument or telemetry failure. For singular erroneous values, and for small gaps of one to two records in length, data were infilled using the average of values either side of the break, or by extrapolating previous values. In the event of longer periods of missing data, the entire record was deemed unusable and was removed from the database. After constructing the 24-hour forecast time-series of the downstream chlorine residual, and lags of up to 48 hours for each parameter, the available modelling data comprised a total of 2 773 hourly records.

7.5.3 Selected input variables

The modified PMIS algorithm developed was applied to select a set of input variables from the 384 candidate variables available for inclusion. The results of the analysis are summarised in Table 7.7, which indicates the PMI corresponding to the most salient variable identified at each iteration. The AIC-based termination criterion resulted in the selection of 10 input variables. Use of the Hampel-test based termination criterion resulted in the selection of the first four input variables, with subsequent variables failing the significance test. An immediate observation is that neither set of input variables includes flowrate, pH, or turbidity. Rather, the selected input sets describe a predominantly auto-regressive time-series structure within the data, with a small contribution from exogenous lags of upstream chlorine and temperature.

Models developed using the input sets selected using PMIS with the AIC-based and Hampel test-based termination criteria are denoted as Models A and B, respectively. As for the Cherry Hills–Brushy Plains WDS case study, additional input variable sets—one consisting of all available parameters at time t only, and the other comprising all available lags $(t, \dots, t-48)$—were considered for comparison purposes. The models corresponding to the these input variable sets are denoted as Models C and D, respectively. The input sets corresponding to all models are summarised in Table 7.8.

Table 7.6: Statistical summary of historical data collected for the Myponga WDS case study.

Parameter	Variable	Min	Max	Ave	St. Dev.	25%-ile	75%-ile	Interquartile Range
Primary free chlorine (mg/L)	C_{WTP}	0.00	6.49	3.17	0.62	3.04	3.48	0.44
Filtered water pH	pH	5.34	14.00	7.19	0.22	7.14	7.25	0.11
Filtered water turbidity (NTU)	Tu	0.00	2.00	0.06	0.07	0.04	0.08	0.04
FWS tank level (%)	H	43.39	100.00	78.61	9.24	72.39	85.87	13.48
FWS free chlorine (mg/L)	C_{FWS}	1.03	2.59	2.01	0.26	1.85	2.22	0.37
FWS outlet flow (ML/d)	Q	6.31	51.54	17.37	7.73	10.42	23.48	13.06
Water temperature (°C)	T	11.03	26.30	16.75	3.51	13.13	19.60	6.47
Downstream free chlorine (mg/L)	C_{WDS}	0.12	1.87	0.75	0.33	0.50	0.97	0.47

Table 7.7: PMIS analysis of input variables for the Myponga WDS case study.

Iteration	Candidate	PMI	Termination
1	$C_{\text{WDS}}(t)$	1.087	
2	$C_{\text{FWS}}(t)$	0.105	
3	$T(t-13)$	0.106	
4	$C_{\text{WDS}}(t-24)$	0.090	Hampel test
5	$C_{\text{WDS}}(t-47)$	0.074	
6	$C_{\text{WDS}}(t-3)$	0.070	
7	$C_{\text{WTP}}(t)$	0.065	
8	$C_{\text{FWS}}(t-17)$	0.062	
9	$C_{\text{WDS}}(t-27)$	0.062	
10	$C_{\text{WDS}}(t-1)$	0.071	AIC

Table 7.8: Summary of input variables selected for GRNN models of the Myponga WDS

Model	# Inputs	Inputs
A	10	$C_{\text{WDS}}(t)$, $C_{\text{FWS}}(t)$, $T(t-13)$, $C_{\text{WDS}}(t-24)$, $C_{\text{WDS}}(t-47)$, $C_{\text{WDS}}(t-3)$, $C_{\text{WTP}}(t)$, $C_{\text{FWS}}(t-17)$, $C_{\text{WDS}}(t-27)$, $C_{\text{WDS}}(t-1)$
B	4	$C_{\text{WDS}}(t)$, $C_{\text{FWS}}(t)$, $T(t-13)$, $C_{\text{WDS}}(t-24)$
C	7	$C_{\text{WTP}}(t)$, $pH(t)$, $Tu(t)$, $C_{\text{FWS}}(t)$, $Q(t)$, $T(t)$, $C_{\text{WDS}}(t)$
D	384	C_{WTP}, pH, Tu, C_{FWS}, Q, T, C_{WDS} $\forall(t,\ldots,t-48)$

Table 7.9: Test performance for 24-hour forecasts of residual chlorine within the Myponga WDS.

Model	RMSE*	MAE*	MRE*	r^{2*}	AIC
	(mg/L)	(mg/L)			
Model A	0.0550 (0.0108)	0.0330 (0.0077)	0.0531 (0.0123)	0.9862 (0.0056)	13.5
Model B	0.0657 (0.0060)	0.0385 (0.0033)	0.0628 (0.0058)	0.9808 (0.0039)	7.6
Model C	0.1027 (0.0149)	0.0610 (0.0101)	0.1000 (0.0181)	0.9525 (0.0136)	14.0
Model D	0.1022 (0.0562)	0.0724 (0.0149)	0.0823 (0.0381)	0.8952 (0.0251)	768.0

*Values in parentheses denote standard deviation.

7.5.4 Model performance

The performance of GRNN models developed using the inputs in Table 7.8 are summarised in Table 7.9 and Table 7.10 for test and validation data, respectively. Each table shows the average error for the ensemble of GRNN models trained on independent data subsets. The standard deviation of results in both test and validation results was low, which indicates the results for each individual GRNN in the ensemble trained on independent samples has low sample variability. A two-tailed t-test indicated that the variance of the model performance had no statistical bearing on the comparison of relative performance based on the mean performances of each model. The results for test and validation data show a high degree of consistency, which confirms the good generalisation performance of the models achieved by the ensemble training.

In terms of accuracy alone, the best performance was obtained by Model A, which used ten input variables selected by PMIS with the AIC-based termination criterion. This model had the lowest average RMSE (0.055 mg/L). However, comparison of the corresponding AIC values indicates that the smaller set of inputs used in Model B resulted in a more efficient model. The results for Model C indicate that the GRNN with no lagged input variables performed poorly, with an error approximately twice that of models A and B. Model D had the worst validation performance, with the highest MAE (0.0724 mg/L) and the lowest r^2 value of 0.8952, which shows that, for real-world data that contain noise, models using a large input variable set can perform more poorly than those with fewer inputs, and that inclusion of superfluous variables can reduce model performance.

Table 7.10: Validation performance for 24-hour forecasts of residual chlorine within the Myponga WDS.

Model	RMSE* (mg/L)	MAE* (mg/L)	MRE*	r^{2*}	AIC
Model A	0.0654 (0.0142)	0.0342 (0.0062)	0.0550 (0.0103)	0.9805 (0.0095)	13.6
Model B	0.0695 (0.0075)	0.0388 (0.0024)	0.0634 (0.0044)	0.9784 (0.0046)	7.7
Model C	0.1159 (0.0185)	0.0633 (0.009)	0.1024 (0.0188)	0.9390 (0.02)	14.1
Model D	0.1061 (0.0595)	0.0746 (0.0133)	0.0844 (0.0391)	0.8879 (0.025)	768.1

*Values in parentheses denote standard deviation.

The ability of Model B to forecast chlorine disinfectant residuals 24 hours in advance is illustrated in Figure 7.5, which shows a portion of the original time-series, $C_{WDS}(t)$, as a solid line, with corresponding test and validation forecasts generated by Model B (for one instance of training, test and validation data) indicated by unfilled and filled markers, respectively. It should be noted that it is necessary to plot the forecasts in this way, as the time-series order of data was not preserved in test and validation data subsets due to the random sampling procedure used for hold-out validation.

Overall, the results obtained are comparable to those reported previously by *Serodes et al.* (2001) and *Bowden et al.* (2006) for similar applications, and support the suitability of sparse ANN models, which utilise a minimum set of input variables, for generating forecasts of residual chlorine within distribution systems. Potential applications include the development of early warning systems that are able to predict fluctuations in downstream water quality in advance, in order to allow operators to make any necessary adjustments to the chlorine dose.

7.6 Discussion

7.6.1 Model parsimony

The results of the case studies presented in this paper support the finding by *Serodes et al.* (2001) that maximising the available information regarding the dynamics of the WDS leads to optimal model performance. The process delays and

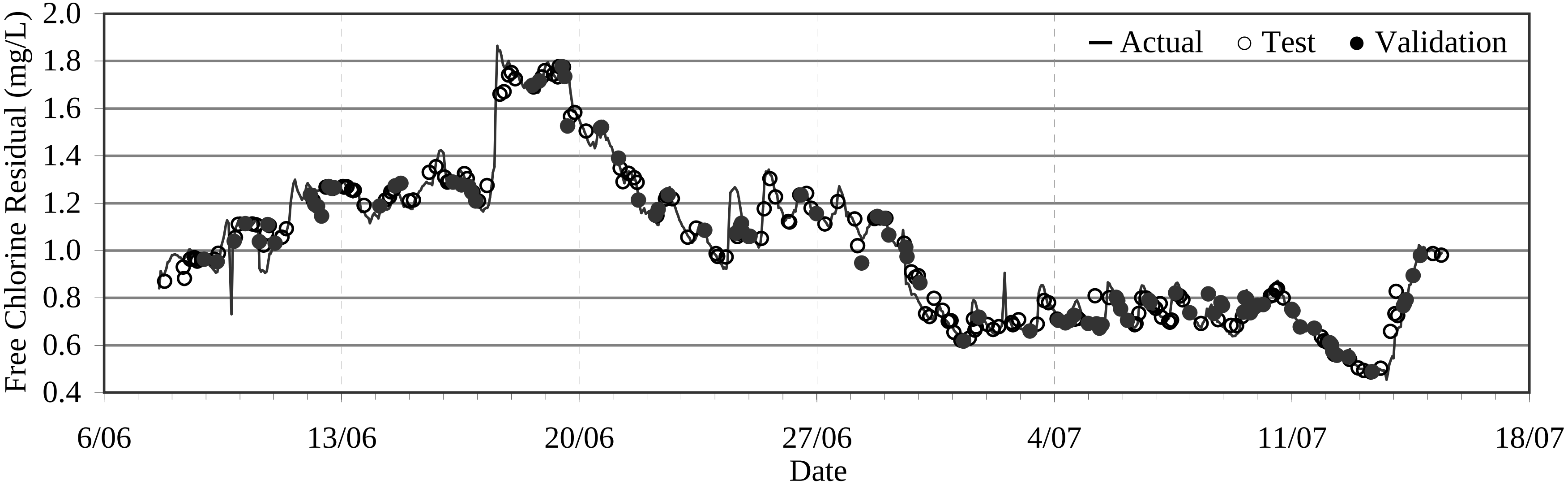


Figure 7.5: 24-hour test and validation forecasts of free chlorine residual generated by Model B for an instance of training, test and validation data.

detention times within a WDS can be long (up to several days), and the ANN needs to contain sufficient input variables to capture the dynamics over this period to generate the best possible forecasts. However, the case studies presented in this paper have also demonstrated that, while the inclusion of dynamic variables is important, there are many redundant and irrelevant parameters that can be excluded from the ANN model without sacrificing forecasting accuracy. Given the importance of selecting as few inputs as possible, there is a clear case for utilising an algorithm such as PMIS during ANN development.

Based on the results, neither of the two PMIS termination criteria used were found to clearly perform better than the other, and the use of each criterion resulted in an accurate model that incorporated a relatively efficient subset of the entire lagging window. Models that were developed with inputs selected using PMIS compared favourably with models that utilised no lagged variables, and those using all lagged variables, although there were differences in each case study between the number of input variables selected. Given the potential size of the lagging window and the large number of candidate input variables, the difference in efficiency will be relatively small in comparison to the overall efficiency gained by the use of PMIS. This study has demonstrated that the criteria are suitable for real-world IVS applications, and supports the results previously presented in *May et al.* (2008a), which were based on synthetic examples that were used to develop and evaluate the novel PMIS termination criteria.

7.6.2 Comparison of developmental frameworks

Current methods for ANN development require trial-and-error procedures to construct an optimal ANN model. In the approach presented in this paper, the IVS procedure yields an optimal model through statistical analysis of the input-output relationships that exist within the data. In the examples given, ANN models were developed for unique water distribution systems using the same approach, without the need for *a priori* expert knowledge or heuristics. Even where the input variables selected are consistent with previous approaches, the methodology is based on analysis of the data, rather than on heuristics, and thus provides a more rigorous basis for the inclusion of input variables.

The importance of a consistent, data analysis-oriented framework for ANN development becomes more apparent when considering the future application of ANN models to more complex water distribution systems. The relatively simple case studies considered thus far involved up to 500 candidate variables. For larger systems, the number of variables to consider could quickly increase to the order of

1 000 as the number of available parameters and the number of lags increase. As the complexity of the water distribution systems under consideration increases, the decision of which variables to include as inputs will become less intuitive, and modellers will find algorithms such as PMIS to be of immense value.

7.6.3 Interpretability of forecasting models

A perceived shortcoming of the ANN modelling approach is that the forecasts generated by an ANN model are somewhat inexplicable. Although an ANN model is able to generate accurate forecasts, *Serodes et al.* (2001) state that *"... owing to its black-box nature, the results obtained cannot be explained."* The lack of interpretability is not surprising, since current methods for ANN development are somewhat holistic in that they do not consider the contribution of individual input variables to the model. However, from the results of the IVS implemented during model development using PMIS, it is apparent that the ANN can provide efficient and accurate predictions based on specific relationships that are identified within the data.

A review of the selected input variables can provide a simple, qualitative analysis of the specific patterns that are identified within the data by the IVS algorithm, and are then able to be used by the ANN to generate predictions. Water distribution systems are known to exhibit a strong periodicity due to diurnal patterns in demand, which are a major contributing factor in observed water quality behaviour (*Polycarpou et al.*, 2002). Periodic behaviour was evident in both case studies, as observed for the chlorine time-series shown in Figs. 7.4 and 7.5. It would appear, based on the input variables selected, that 24-hour cyclic behaviour is an important component within the data. For example, consider the input variables selected in the second case study, which included past values of the output at time t, $t - 24$, and $t - 47$. The result is consistent with the notion of *tendency* in periodic, or oscillatory systems that has been defined elsewhere for similar forecasting applications where trends exist over homologous observations within the period of oscillation (*Santos et al.*, 2005). The selection of sequences of endogenous variables (e.g. C_{WDS} at time t, $t - 1$, and $t - 3$) suggests that the current state, and immediate rate of change of the system, are also necessary for prediction. Interestingly, a similar pairing of endogenous variables at $t - 24$ and $t - 27$ was also selected as input variables, which suggests further reinforcement of short-term behaviour due to the 24-hour periodicity of the system.

Further insight regarding the mechanisms by which the forecasts are generated could be gained by a more quantitative method, such as sensitivity analysis of the models post-development. However, information regarding the importance of each input variable can be inferred based on the statistical measurement of input-output strength that forms the basis of the IVS approach. A measure of relative importance (RI) has been proposed based on analysis of the PMI for individual input variables (*Soofi and Retzer*, 2003), and which has been found to give results that are in good agreement with methods based on analysis of the trained ANN (*Kingston*, 2006). The RI is determined directly from the PMI by the expression

$$\mathrm{RI}_i = \frac{I'_{C_i Y \cdot X}}{\sum_{j=1}^{k} I'_{C_j Y \cdot X}} \times 100\%, \tag{7.11}$$

where $I'_{C_i Y \cdot X}$ is the PMI between candidate C_i and Y conditional of selected input variable set X, which is estimated at each iteration of the PMIS algorithm.

The relative importance measures the relative contribution provided by each input variable, which indicates which relationships are likely to be predominantly used by the ANN model. The cost of monitoring and collecting data for a large number of variables can be a significant factor when evaluating the cost-to-benefit ratio of model development. A quantitative measure of RI is therefore useful in estimating the expected trade-off between model accuracy and the number of input variables used.

Based on the selected input variables in Table 7.7, it is apparent that the ANN developed is predominantly an auto-regressive time-series model, since it is dominated by endogenous lagged variables. The model is a highly accurate representation of the time-dynamics of water quality within the WDS, and as such can provide an early indication of downward trends in free chlorine residual that may warrant corrective action by the WTP operators. However, the PMI of applied chlorine dose, $C_{WTP}(t)$ is relatively low, as is the PMI for residual at the FWS outlet $C_{FWS}(t)$, indicating that only a weak statistical relationship was established between upstream and downstream chlorine. In the case of the data collected for the Myponga WDS, it is evident that the residual free chlorine data are simply either too noisy, or the data contain insufficient variance to determine a relationship between applied dose and future downstream residual. This is likely, given the expected dampening of chlorine residual fluctuations over the span of the trunk main. Furthermore, it was not permissible to manipulate the chlorine injection rate, which meant that adjustments to the applied chlorine dose were typically small and infrequent, which resulted in the relatively low variability of chlorine residual within the system. It would be difficult to recommend use of the GRNN model to directly determine the influence of changes in applied dose. The development of control-oriented ANN models will need to consider other factors such as limited observability of the water distribution system (i.e. low variability of chlorination), noise, and process delays. In particular, due to the restricted scope for experimentation with disinfection parameters, the collection of operational data with sufficient variability for modelling chlorine disinfection presents a key area for future efforts, which should aim to define appropriate experimental protocols for undertaking WDS identification in a manner that ensures water quality is not compromised.

It is worth noting that, in view of the additional information regarding the significance of input variables, a forecasting model may not be suitable for some model-based control applications; which presents an important limitation in terms of potential applications. Based on the selected input variables in Table 7.7, it is apparent that the ANN developed is predominantly an auto-regressive time-series model, since it is dominated by endogenous lagged variables. The model is a highly accurately representation of the time-dynamics of water quality within the WDS, which may be of some use to operators, although it would be difficult to recommend the use this particular model to directly determine the influence of changes in applied dose. This results highlights that the development of analysis and diagnostic tools that can to assess the functionality of ANN models is also important, since inappropriate assumptions regarding the utility of black-box models could be misleading.

7.7 Conclusions

This paper has reported the application of the modified PMIS algorithm, introduced by *May et al.* (2008a), to the development of ANN models for forecasting disinfectant residual within water distribution systems. The algorithm identifies an optimal subset of candidate ANN input variables by analysing partial mutual information (PMI). The performance of ANN models developed using this approach was found to be favourable in comparison with those developed using a current methodology reported in the literature, which is based on expert knowledge and trial-and-error. The primary benefit of the approach is that it identifies the minimum number of ANN input variables required to forecast disinfectant residual, without loss of prediction accuracy.

In terms of ease of development, the selection of input variables based on analysis of the data, rather than application of specific heuristics, has considerable appeal. Studies presented thus far have considered IVS for ANN models of simple distribution systems. In this research, the application of a consistent, analysis-oriented approach to IVS has been shown to yield highly accurate ANN models for two different water distribution systems. Consequently, as modellers seek to apply ANN models to increasingly complex water distribution systems, or begin to consider additional water quality variables, the approach presented in this paper offers a more systematic way of selecting input variables without the loss of accuracy.

Finally, the identification of a set of specific input variables goes some way to increasing the transparency of the ANN modelling methodology. This is because the subset of variables selected during PMIS provides an indication of the specific relationships within the data that are necessary for the ANN to learn in order to generate predictions. As a result, the ANNs produced are simpler and more easily interpreted. Furthermore, the statistical analysis of input-output relationships that form part of the IVS procedure provides additional insight into the relative importance of each of the ANN input variables. Understanding the importance of variables is extremely useful for assessing the relative cost-benefit of establishing and maintaining monitoring systems for generating modelling data.

Acknowledgements

This research was aided by the use of facilities provided by the South Australian Partnership for Advanced Computing (http://sapac.edu.au) and was partially funded by United Water International Pty Ltd and the Cooperative Research Centre for Water Quality and Treatment. The authors would also like to thank the anonymous reviewers for their comments, which have substantially improved the quality of this paper.

Chapter 8

Data Splitting for Artificial Neural Networks Using SOM-based Stratified Sampling

Publication 4

Publication Details

This work has been published within the journal *Neural Networks* as the following article:

> May, R. J., H. R. Maier, G. C. Dandy, Data Splitting for Artificial Neural Networks Using SOM-based Stratified Sampling, *Neural Networks*, 23, 283–294, 2010.

Although the manuscript has been reformatted in accordance University guidelines, and sections have been renumbered for inclusion within this thesis, the paper is otherwise presented herein as submitted.

Statement of Authorship

May, R. J. (Candidate)
Literature review, conceptual development, experimental design, software development, analysis and interpretation of results, manuscript preparation and corresponding author.

Signed: .... Date: ....

Maier, H. R.
Research supervision and review of manuscript.

Signed: .... Date: ....

Dandy, G. C.
Research supervision and review of manuscript.

Signed: .... Date: ....

Abstract

Data splitting is an important consideration during artificial neural network (ANN) development where hold-out cross-validation is commonly employed to ensure generalisation. Even for a moderate sample size, the sampling methodology used for data splitting can have a significant effect on the quality of the subsets used for training, testing and validating an ANN. Poor subset selection can result in an inaccurate and highly variable model performance, however the choice of sampling methodology is rarely given due consideration by ANN modellers.

This paper provides a comprehensive review of the various sampling algorithms for data splitting, and the quality of subsets that are obtained. An algorithm for stratified sampling, based on the self-organising map (SOM) is then developed, with several guidelines for implementing the approach to minimise bias and variance in the datasets. Results for an example problem show that a stratified sampling technique with Neyman allocation consistently yields high quality samples and can be used with greater confidence than other sampling techniques, especially in the case of non-uniform multivariate datasets. Increased confidence in the sampling is of paramount importance, since the hold-out sampling is performed only once in development.

8.1 Introduction

Statistical models of all types are invariably built using a finite set of data. It is rare that data collected through observation of a process will be noise-free, and available data are likely to contain a small proportion of features that are not representative of the underlying process. Generalisation is therefore a central issue for the development of all statistical models. Generalisation refers to the ability of a model to accurately represent the underlying data generation process, rather than the idiosyncratic features of the training data. The latter case is referred to as *over-fitting*, because it is characterised by a high goodness-of-fit to the training data, yet poor accuracy when applied to previously unseen data. Despite their many advantages over conventional statistical models, artificial neural networks (ANNs) are particularly susceptible to over-fitting due to the complexity of the model architecture (i.e. the number estimated parameters) relative to the number of training data. In statistical learning theory, this is referred to as the *bias/variance dilemma*, since there is a trade-off between minimising the model error (bias) and minimising the dependence of the estimated parameters on the training data (variance) (*Geman et al.*, 1992).

Methods commonly used during the development of statistical models to ensure good generalisation include hold-out cross-validation, k-fold cross-validation, ensemble training, and Bayesian regularisation (*Sarle*, 1997). In ANN applications, the hold-out is most commonly employed, and is synonymous with *stop-training* or *early stopping*. In this approach, a subset of data is reserved to periodically test the performance of the network during training. Training is stopped when the test error reaches an optimum value, as further training will result in over-fitting, and hence ensures a generalised fit. Furthermore, when model selection is employed to compare alternative models or optimise ANN architectures, the models can potentially be optimistically biased towards the test data. In order to avoid testing bias, a second hold-out is required for validating the optimal ANN model (*Maier and Dandy*, 2000).

Regardless of the number of data subsets required, the issue for modellers is that the hold-out of data itself can prove to be yet another source of bias and variance. If the data subsets are selected inappropriately, then training, test and validation data may not be equally representative of the problem domain, and will generate inaccurate test or validation performance. Variation in test and validation error may be observed for repeated instances of sampling, which creates uncertainty regarding the model performance that is gauged based on a single instance of training, test and validation data. The uncertainty due to sampling variance may be significantly greater than other sources of model uncertainty such as network initialisation, training and architecture (*LeBaron and Weigend*, 1998).

On the subject of selecting training data for ANN development, *Sarle* (1997) remarks only that: *"Methods for the selection of training data can be found in statistical textbooks."* However, ANN practitioners working within the context of machine learning are not necessarily conversant with statistical sampling theory. The result is that in many applications, random sampling or some other arbitrary method is used. The importance of data splitting is ignored or understated, with more attention given to the model architecture and learning algorithm. The potential bias and variance in model performance due to sampling is often never tested, and data splitting is generally performed only once during ANN development. The final point here highlights the importance of understanding the implications of the data splitting technique on subsequent model development, given that the assessment of model performance will be dependent on a single data split.

This paper compares data splitting algorithms for ANN development in terms of their relative bias and variance. In particular, a data splitting method based on stratified sampling of the self-organizing map (SOM) is introduced to reduce the bias and variance of ANN performance, relative to other approaches. The remainder of the paper is structured as follows: Section 8.2 briefly reviews the different approaches for data splitting that have been applied to ANN development. Section 8.3 discusses the issues surrounding the implementation of the SOM-based data splitting approach. Details of the experimental study are given in Section 8.4 and results are given in Section 8.5. Finally, concluding remarks are given in Section 8.7.

8.2 Data Splitting Methods

Data splitting for ANN development is essentially a sampling problem where, given a database $\mathcal{D}$ comprising N data, the goal is to sample the data into disjoint subsets $\mathcal{T}$, $\mathcal{S}$ and $\mathcal{V}$ of size $N_\mathcal{T}$, $N_\mathcal{S}$ and $N_\mathcal{V}$, for training, testing and validating, respectively. Within ANN literature, this task has been performed using many different approaches, each with their advantages and disadvantages.

Simple random sampling (SRS) is the most common method for data splitting in ANN development, where data are selected with uniform probability, which is determined as

$$p(x \in S_{\mathcal{T}}) = \frac{N_{\mathcal{S}}}{N}, \tag{8.1}$$

and similarly for $x \in S_{\mathcal{S}}$ and $x \in S_{\mathcal{V}}$. Simple random sampling is easy to perform, and can be efficiently implemented in just a single pass over the data using algorithms such as Knuth's algorithm (*Knuth*, 1997). However, the problem with this approach is that there is a chance that the splitting of data suffers from variance, or bias, especially when the data are non-uniformly distributed (*Tourassi et al.*, 2001).

More arbitrary sampling methods include the splitting of datasets according to discrete blocks. Sampling methods such as these are traditionally classed as *convenience* sampling, for which the introduction of unknown bias is a common criticism. For example, splitting data according to discrete time intervals is common in time-series model development (*Bowden et al.*, 2002). However, unless the time-series is stationary, the presence of long-term trends within the data, or differences in the features and events observed during the different time intervals, can lead to unrepresentative training and testing data, which results in poor model performance (*Bowden et al.*, 2002).

Simple trial-and-error methods have been proposed on the basis that equally representative datasets will have similar statistics. More sophisticated approaches have utilised an optimisation loop to automate the search through the combination of potential splits (*Reeves and Taylor*, 1998; *Bowden et al.*, 2002, 2006). Various approaches have aimed to minimise the difference in statistics such as the mean, μ, and standard deviation, σ (*Bowden et al.*, 2002; *Shahin et al.*, 2004); or, have used the Kolmogorov-Smirnov statistic to match the distributions of each variable across the sampled datasets (*Bowden et al.*, 2005). To the authors' knowledge, the validity of this assumption has yet to be thoroughly tested. However, global statistics of subsets will not indicate an unbalanced representation of local features of the database within each, which might then require extrapolation; and ANNs are known to perform poorly in these circumstances.

CADEX, or Kennard-Stone sampling, (*Kennard and Stone*, 1969) is one of the earliest algorithms designed for data splitting. The approach iteratively draws samples based on distance, selecting points farthest away from those already included in the sample, and ensures maximum coverage of the data. An improved version called DUPLEX was proposed by *Snee* (1977), which is used widely in the field of chemometrics, including several ANN applications (*Despagne and Massart*, 1998; *Sprevak et al.*, 2004). However, the computational complexity of this algorithm may prohibit its use on large datasets.

Systematic sampling is another deterministic approach in which every k^{th} observation is sampled. If the data are ordered in some way, this implicitly generates a stratified sample, with stratification on the ordinal variable. One approach is to sort the data along the output variable dimension to obtain a representative sample of the output variable distribution (*Baxter et al.*, 2000). This approach is easy to implement, as it assumes that the output variable can be mapped to a unique input state. However, this assumption may not hold in multivariate datasets where multiple input states might give rise to the same output, where the method cannot ensure that representative input-output combinations will be sampled, since only the output variable is considered.

Stratified sampling partitions the data into H homogeneous groups (or, strata) of size N_h, and data are sampled from within each stratum (*Cochran*, 1977). The partitioning of the data forces sampling to be distributed throughout all regions of the input-output space, and ensures that adequate representation of input-output tuplets can be achieved. For multi-variate data it is convenient to use partitioning or clustering algorithms to generate the strata (*Mulvey*, 1983). *Gill et al.* (2004) refer to this as cluster-based stratified sampling (CBSS). Several examples of data splitting have been described using different clustering algorithms, including k-means clustering, the self-organizing map (SOM) (*Kohonen*, 1995) and fuzzy c-means clustering (*Kaufman and Rousseeuw*, 1990). *Svozil et al.* (1995), *Daszykowski et al.* (2002) and *Bowden et al.* (2002) applied a partitioning of data based on the self-organizing map prior to sampling. The methodology has since been adopted in several similar ANN applications to water resources modelling (*Anctil and Lauzon*, 2004; *Zhang et al.*, 2004a; *Kingston*, 2006). *Shahin et al.* (2004) used fuzzy c-means clustering to partition the data for ANN model development, where the membership values were used to guide sampling.

The approach based on the SOM appears to be an attractive method for data splitting, since it is a relatively robust clustering algorithm (*de Bodt et al.*, 2002). Given that it is another class of ANN, it also may already be relatively familiar to ANN modellers. However, despite being used in several examples within the literature, there are marked differences in the specific manner in which this type of data splitting is performed. In particular, the best method for selecting the SOM size, and the manner in which samples are selected from SOM units is unclear (*Daszykowski et al.*, 2002). Furthermore, the approach has yet to be compared to existing approaches, such as DUPLEX. More importantly, no study so far has considered the implications of the approach in terms of bias and variance of model performance, and so the relative benefits of the approach for ANN development have not yet been fully assessed.

8.3 SOM-based Stratified Sampling

The self-organizing map (SOM) (*Kohonen*, 1995) is a class of unsupervised ANN that is represented as an array of p-dimensional vectors. The SOM can generate a partitioning of data by learning the optimal distribution of the weight vectors. This forms the basis of SOM-based stratified sampling (SBSS), where the SOM is used to generate a partitioning of data, and then samples are drawn from the SOM partitions. Conceptually, SBSS provides a convenient method for implementing stratified sampling. However, as mentioned there are currently no guidelines for how to best implement the approach. In particular, there are several key considerations (*Kpedekpo*, 1973):

1. choice of stratified variables;
2. number of strata;
3. location of strata boundaries; and
4. allocation of samples.

The first three considerations are essentially analogous to the following decisions that are normally considered when partitioning data using the SOM: choosing the variables to cluster (input variables to the SOM), the number of SOM map units, and how the SOM can optimally determine clusters. The final consideration (sample allocation) is unique to the data splitting application of the SOM, however, all four are interdependent and will influence the quality of the sampling.

8.3.1 Choice of Variables

Gill et al. (2004) suggest that stratification on the most relevant variable can yield better results than clustering on all results, and refer to this technique as induction based stratified sampling (IBSS). However, this does not account for the situation of multiple variables of equal importance, although we can suggest that in this case, all informative variables should be used. Extending this notion to the most general case, it can be simply said that the key consideration here is

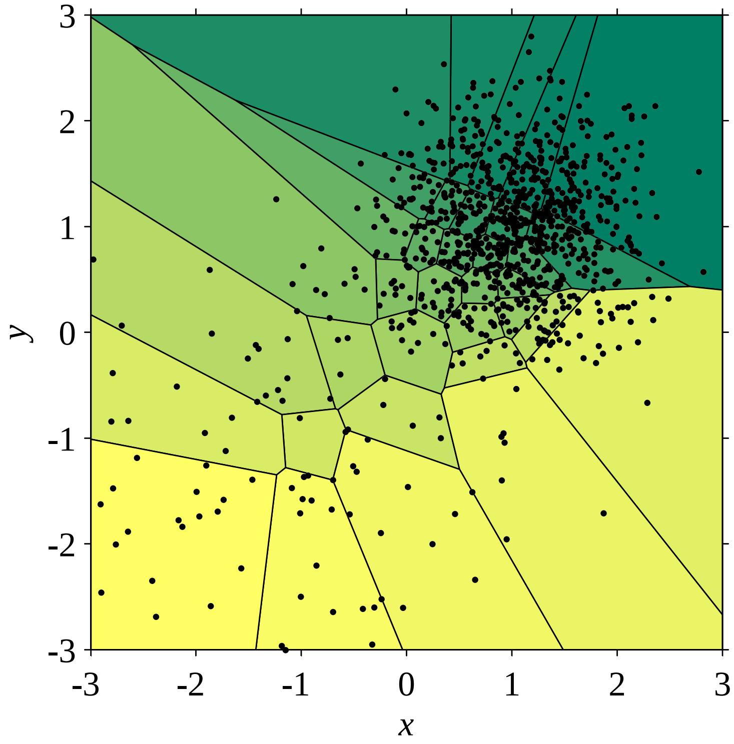


Figure 8.1: Partitioning of data by an 7 × 5 SOM, where the Voronoi regions denote the boundaries of partitions created by the weights of the trained SOM. The shading of the regions corresponds to rows of the SOM grid, indicating the alignment of the map through the data.

to avoid clustering on irrelevant or redundant variables. To this end, we observe that good ANN model development adopts an input variable selection (IVS) step that precedes the data splitting (*May et al.*, 2008a, 2009a). The result is that the set of available data will contain only input variables relevant to the output variable, and will be free of noise variables. Consequently, partitioning over all dimensions should discriminate all distinct regions of input-output tuplets.

8.3.2 Location of Strata Boundaries

The use of the SOM provides a robust method for stratification, since the learning algorithm determines the optimal positioning of the SOM prototype vectors throughout the data space. The boundaries of strata are then formed by the boundaries of the Voronoi regions that are formed by the partitioning, as shown in Figure 8.1. The reliability of SOM partitioning is influenced in varying degrees by the SOM parameters i.e. map size, learning rate, number of training iterations, of which map size is the predominant parameter (*de Bodt et al.*, 2002).

8.3.3 Sample Allocation

Sample allocation refers to the sampling within each stratum, and is an important aspect of SBSS in terms of selecting data for ANN development. In past applications of SBSS, the sample allocation has varied. *Svozil et al.* (1995), *Daszykowski et al.* (2002) and *Bowden et al.* (2002) draw single samples from each cell for training, test and validation, although these are based on different grid sizes. Alternatively, *Kingston* (2006) randomly samples all data within each partition in proportion to the desired sample sizes for training, testing and validation, so that all of the available data are used. It remains unclear which, if any, is the most appropriate approach to take.

In stratified random sampling, the sampling within strata is usually uniform random sampling, and an allocation rule identifies the number of samples drawn per stratum, which is referred to as the *quota*. Three basic rules can be considered for determining the sample quota (*Kpedekpo*, 1973; *Cochran*, 1977): equal allocation, proportional allocation, and Neyman allocation.

Equal allocation

Equal allocation is the simplest way to allocate samples, and takes an equal number of points from within each stratum. The number of samples drawn per stratum, n_h is given as

$$n_h = \frac{n}{H}, \tag{8.2}$$

where n is the required sample size, and H is the number of strata. The allocation rule implies that $N_h \geq n_h$, otherwise the rule will break down, and the SOM map size may need to be restricted to ensure that sufficient data are available in each cluster to draw data for training, testing and validation. The method used by *Svozil et al.* (1995); *Daszykowski et al.* (2002) and *Bowden et al.* (2002) is essentially the case of equal allocation where $n_h = 1$.

Proportional allocation

The allocation of samples from within strata can also be determined based on the size of individual strata, N_h, to yield proportional allocation. In this case, the number of samples n_h to be taken from stratum h can be determined according to

$$n_h = \frac{N_h}{\sum_{j=1}^{H} N_j} n, \tag{8.3}$$

which results in the overall selection of n samples.

Neyman allocation

Neyman allocation considers both the size of the stratum, N_h, and the intra-stratum standard deviation σ_h. The Neyman allocation for stratum h is determined according to

$$n_h = \frac{N_h \sigma_h}{\sum_{j=1}^{H} N_j \sigma_j} n, \tag{8.4}$$

where σ_j is the intra-stratum standard deviation. Here, the sample allocation is increased for strata that are either large, or have increased variance. Neyman allocation yields an optimal sample when used to draw a stratified sample for the estimation of conventional statistics (*Cochran*, 1977). It should be noted that σ_j conventionally refers to the within-stratum standard deviation of the variable for which statistical estimates are later generated, since this defines the optimality of the sample allocation rule. However, in this study the standard deviation is based on the multivariate form,

$$\sigma = \sqrt{\sigma_{x_1}^2 + \cdots + \sigma_{x_p}^2 + \sigma_y^2} \tag{8.5}$$

where $\sigma_{x_i}^2$ is the intra-stratum variance of component x_i, and σ_y^2 is the variance of the output variable, y. The multivariate standard deviation describes the within stratum variability with respect to input-output tuplets, which is for sampling data for regression. In this case the sampling rate is expected to be greater where there is greater variance in either the inputs, output, or both.

Consider the partitioning of data generated by a mixture model, as shown in Figure 8.1, which shows the Voronoi tesselation of the data space due to a SOM partitioning. The Voronoi regions each define the region in $\Re^d$ where points inside lie closest to a given prototype vector, c_i. Here, the data are non-uniformly distributed, and as can be seen by the Voronoi regions, the SOM clusters are non-uniformly distributed. In fact, due to the behaviour of the SOM learning algorithm, the distribution of SOM partitions approximates that of the data, and the SOM does not necessarily generate clusters of equal size or width. Equal allocation and proportional allocation may potentially overlook this aspect of the SOM clustering, and under-sample some regions that would ideally be allocated a greater sample quota. On the other hand, Neyman allocation allows for the sample quota to be adjusted according to the width of the clusters. Neyman allocation will increase the sample quota in sparse map units, and this would result in a similar sampling approach to density biased sampling with a negative bias.

8.3.4 Number of Strata

Determining the number of strata is analogous to the problem of determining the size of the SOM, which is usually an $m \times n$ grid. Selecting the most appropriate size SOM is generally a non-trivial task for which no conclusive rule has been determined, and certainly different approaches have been taken in various applications of SBSS. For example, *Daszykowski et al.* (2002) used a SOM with $\sim N$ units, and selected one datum each per cluster. *Bowden et al.* (2002) used enough map units to capture all clusters, and reported successful clustering using a 10×10 map. *Mulvey* (1983) observes that optimal clustering can yield an optimal sample. In clustering applications, two approaches for determining the optimal number of SOM units are: cluster validity (*Kaufman and Rousseeuw*, 1990; *Halkidi et al.*, 2001), and a heuristic rule size (*Vesanto*, 1999).

Cluster Validity

Cluster validity describes the quantitative evaluation of the output of a clustering algorithm, based on a cluster validity index (CVI) that defines an expression for the relative intra-cluster similarity (or *cluster compactness*) and inter-cluster dissimilarity (or *cluster separation*). Good clustering creates clusters that are compact and well separated from each other (*Kaufman and Rousseeuw*, 1990). Numerous CVIs have been described (see *Gunter and Bunke* (2003), *Halkidi et al.* (2001) for examples), but are generally based on estimation of the ratio of cluster compactness to cluster separation. Compactness and separation are invariably measured by some analysis of distances between points within the same cluster, and distances between points in different clusters.

The application of cluster validity to stratified sampling of ANN data subsets has been considered in only a few examples. *Shahin et al.* (2004) demonstrated the application of the silhouette coefficient to optimising the number of fuzzy c-means clusters. *Kingston* (2006) similarly extended the approach described by *Bowden et al.* (2002) by optimising the SOM dimensions based on the silhouette coefficient (*Kaufman and Rousseeuw*, 1990), but also considered the quantisation error and the number of singleton clusters (clusters containing a single datum).

The quantisation error (QE) is the basic measure of how accurately the prototype vectors (i.e. SOM weights) represent the data during and after training. The QE is defined as

$$\mathrm{QE}(k) = \frac{1}{n}\sum_{i=1}^{k}\sum_{x\in C_i}\|x - c_i\|^2, \tag{8.6}$$

which is the MSE between data points in cluster C_i and their respective prototype vector, c_i. The QE is dependent on the number of map units and the neighbourhood size during learning, and a small QE can be achieved by using a small neighbourhood during training (i.e. tuning) and increasing the number of map units. The latter is more or less intuitive, and potentially a QE of zero could be achieved by setting $k = n$, (i.e. letting $C = X$). However, generally it is assumed that $k << n$, otherwise there is little to be gained by the clustering. The limitation of QE for assessing cluster validity is that it only measures cluster compactness, and not cluster separation.

The silhouette coefficient, S, (*Kaufman and Rousseeuw*, 1990) measures the degree of membership of individual points to their respective clusters. S is defined as

$$S(k) = \frac{1}{n}\sum_{i=1}^{n} s_i, \tag{8.7}$$

which is the average silhouette for a given partitioning into a set of k clusters

Table 8.1: Guidelines for interpretation of the silhouette coefficient (*Kaufman and Rousseeuw*, 1990)

$S(k)$	Interpretation
0.70–1.00	Strongly clustered data
0.50–0.70	Reasonably clustered data
0.25–0.50	Only weakly clustered data. Another clustering method might need to be considered.
$<$ 0.25	No structure within the data (i.e. are unclustered)

$C_1, \ldots, C_k$. The silhouette $s_i(k)$, is determined for each $x_i \in C_i$ as

$$s_i = 1 - \frac{b_i - a_i}{\max\{a_i, b_i\}} \tag{8.8}$$

where a_i and b_i are given as

$$a_i = \frac{1}{|C_i|} \sum_{\substack{x_j \in C_i \\ x_j \neq x}} \|x_i - x_j\|, \text{ and} \tag{8.9}$$

$$b_i = \min_{l \neq i} \frac{1}{|C_l|} \sum_{x_j \in C_l} \|x_i - x_j\| \tag{8.10}$$

Here, a_i is the average intra-cluster distance from point x_i, b_i is the minimum average inter-cluster distance, and $|C_i|$ is the size (number of objects) of the i^{th} cluster. The silhouette is bounded on $(-1, 1)$, where $s_i = 1$ indicates that a point x is, on average, much closer to points within the same cluster, than points within the closest neighbouring cluster, C_l. Note that for the special case of singletons, $s_i = 0$. The optimal number of clusters is determined by maximising the silhouette coefficient. *Kaufman and Rousseeuw* (1990) provide some guidelines for interpreting the silhouette coefficient for a data partitioning, which are given in Table 8.1.

The difficulty in applying a CVI is that modelling datasets may not necessarily be clustered, in which case it is unclear how useful the analysis is, or how optimal the sampling will be. The approach has not been rigorously assessed, in terms of minimising bias and variance of the sampling. Although defining optimal clusters can optimise sampling bias and variance of parametric estimates of statistics, the same has not been demonstrated for ANN model validation.

Heuristic SOM Grid Size Formula

Vesanto (1999) suggests a heuristic rule for determining the size of the SOM grid based on the number of samples to be clustered on the map. In this case, the grid size is determined as

$$k = \beta n^{0.54} \tag{8.11}$$

Values of 0.2, 1 and 5 are used for the constant β, which correspond to a small, normal and a large SOM, respectively (*Vesanto*, 1999). Using this heuristic formula, the grid size can be efficiently determined without trial and error. The SOM algorithm also remains scalable, since the number of map units is proportional to $\sqrt{n}$ and reasonable computation times can be maintained for large datasets.

The dimensions of the grid can also affect the quality of the mapping that is achieved by the SOM. It has been observed that an $r \times c$ SOM, with one side greater in length than the other, is better than a square $r \times r$ SOM, since the former is more easily able to align with the training data, which may be distributed along a dominant axis. Given some ratio of the SOM dimensions, γ, it is possible to specify the dimensions of the SOM in terms of the number of rows, by considering that $r = \gamma c$. Since $k = rc$, the number of map units can be therefore be written, in terms of r, as

$$k = \frac{r^2}{\gamma}, \tag{8.12}$$

which can then be substituted into (8.11) to give the number of SOM rows as

$$r = \sqrt{\gamma \beta n^{0.54}}. \tag{8.13}$$

In comparison to the application of a CVI, this heuristic rule is a much simpler and more convenient way to specify the grid size, and does not require the trial-and-error evaluation of clusterings with a potentially large number of SOMs. However, unlike CVIs, the rule in (8.11) has no underpinning theoretical basis and the suitability of this rule for ANN sampling has yet to be determined.

8.3.5 Proposed SBSS Algorithm

An overall methodology for the implementation of SBSS is proposed in this paper, based on the considerations discussed in the previous sections. The SBSS algorithm proceeds as follows:

SOM clustering:

i. Specify SOM map dimensions r and c using (8.13), and $r/c = 1.6$.

ii. Randomly initialise SOM and train on D, using the learning parameters in Table 8.2.

iii. Cluster dataset $\mathcal{D}$ onto the trained SOM.

Sampling:

iv. For each cluster $C^{(m)}$,

v. Calculate the standard deviation $\sigma^{(m)}$ from (8.5).

vi. Determine training quota $n_{\mathcal{T}}^{(m)}$ and test quota $n_{\mathcal{S}}^{(m)}$ using the allocation rule in (8.4).

vii. Randomly sample data without replacement into $\mathcal{T}$ and $\mathcal{S}$.

viii. Allocate remaining data to $\mathcal{V}$.

Here, a rectangular SOM is used to perform the partitioning, and the conventional SOM learning algorithm adopted, with a short global ordering phase, followed by a longer tuning phase (see *Kohonen* (1995) for details). The Neyman allocation rule is used so that the number of data selected from within each cluster is determined based on their size $\left|C^{(m)}\right|$ and spread $\sigma^{(m)}$. Alternatively, the approach could be implemented using different allocation rules (single or proportional) to determine the number of data to be sampled.

8.4 Experimental Study Design

The purpose of the experimental study was to determine the most appropriate methodology for implementing SBSS, given that several different approaches to SBSS have been described. The two key considerations were the size of the SOM, and the sample allocation technique. The influence of these parameters was investigated based on the estimation of bias/variance of an ANN regression using

Table 8.2: SOM parameters for implementing SBSS

Parameter	Ordering	Tuning
Initial learning rate	0.9	0.01
Initial neighbourhood size	r	1
Neighbourhood function	Gaussian	Gaussian
Epochs	2	20
Decay function	Linear	Linear

independent data splits. Additionally, the study aimed to provide a comparison between the SOM-based approach and several sampling algorithms within the literature, and to benchmark them against simple random sampling. Finally, the impact of sampling fraction, that is, the overall proportion of data sampled for training and testing, was also taken into consideration, and experiments were undertaken using 40% and 80% of the data for this purpose.

8.4.1 Datasets

The Friedman regression function (8.14) was used as the basis for assessing the performance of ANN models developed using each data splitting algorithm. The function is given as

$$y = 5\left(2\sin(\pi x_1 x_2) + 4(x_3 - 0.5)^2 + 2x_4 + x_5\right) + \varepsilon \tag{8.14}$$

where ε is Gaussian noise $\sim N(0, 0.8)$. The Friedman function provides a suitable test case for regression applications as it is a well-known function that is a suitably difficult function to approximate, and has a high-dimensional input space. The use of synthetic data was useful for controlling the characteristics of the input domain, such as skewness, noise and correlation between input variables, and number of data.

Datasets of 1 000 observations were generated by independently sampling x_i from an input distribution, and calculating the Friedman function value at each x. Three datasets (Dataset I, II and III) were generated, where the distribution of input variables was varied for each dataset, in order to generate input distributions that represented differing degrees of skewness. Dataset I was a mixture of two Gaussian clusters, generated by sampling 90% of data from the distribution $x_i \sim N(1, 0.6)$, and 10% of data from the distribution $x_i \sim N(-1, 0.6)$. The distribution of data in Dataset I was representative of naturally occurring datasets that are highly skewed. Dataset II was generated by sampling inputs drawn from the $N(0, 0.8)$ distribution, representing a case of more moderate skewness than Dataset I. Dataset III was generated by sampling the input domain using a uniform distribution, $x_i \sim U(-3, 3)$.

Datasets IV, V and VI denote three datasets, corresponding to Datasets I, II and II, respectively; but with correlated input variables. In each case, data were generated by sampling x_1 as before, but generating all remaining input variables x_2 to x_5 according to $x_i = 0.9x_1 + \varepsilon$, where ε is Gaussian noise $\sim N(0, 0.8)$.

8.4.2 Bias and Variance Estimation

The quality of the sampling method was determined by estimating the bias and variance of the ANN error from M bootstrap instances of training, test and validation data samples. The method for estimating the bias and variance is based on the hold-out validation error for M models developed using independently sampled training, test and validation data. In this study, $M = 100$. Given the network MSE, the bias due to the sample is determined as (*Twomey and Smith*, 1998; *Tong and Liu*, 2005)

$$E(\text{MSE}) = \frac{1}{M} \sum_{m=1}^{M} \text{MSE}_m, \tag{8.15}$$

which is the expected error of the model determined for M bootstrap experiments, and is indicative of the representativeness of the sample that is obtained. The sensitivity of model performance to the sample is similarly determined based on the variance of the error, which is given by

$$V(\text{MSE}) = \frac{1}{M-1} \sum_{m=1}^{M} (\text{MSE}_m - E(\text{MSE}))^2. \tag{8.16}$$

Unlike the traditional bias/variance dilemma in ANN training, there is no real trade-off and a sampling algorithm may produce both a low bias and low variance. Such a sampling algorithm would consistently draw data in each set that are representative of the problem domain.

8.4.3 Neural Network Training

The generalized regression class of neural network (GRNN) (*Specht*, 1991) was used to perform the regression. Each neural network was trained by optimising the bandwidth, h, of the Gaussian kernel that is centered at each training pattern. The optimisation is a fast, one-dimensional solution named Brent's algorithm (*Press et al.*, 1992). The GRNN provided a way of training many networks relatively quickly, without having to optimise the model architecture. The objective cost function was the test-set error (MSE) minimisation.

8.4.4 Data Splitting Algorithms

In this comparative study, the three variants of SBSS were implemented, as described in Section 8.3.5, each using different sample allocation rules: single, proportional and Neyman allocation, which are denoted as SBSS(S), SBSS(P) and SBSS(N), respectively. Random initialisation of the SOM was also a potential source of variation in SBSS, since it can potentially lead to variability in the partitioning of data that could, in turn, affect the sample that is drawn. Consequently, in order to quantify how much variability the SOM contributes to the overall variability of SBSS, a variation of bootstrap analysis was undertaken by performing the SOM partitioning only once, and independently drawing random samples from the partitioned data to train each GRNN model. These cases are denoted as SBSS(S)*, SBSS(P)*, and SBSS(N)*. Finally, SBSS was implemented on a large $N \times N$ map with a single sample drawn from each, following the methodology in *Daszykowski et al.* (2002), and this is denoted as SBSS(SL).

DUPLEX

The deterministic DUPLEX algorithm was implemented for comparison with the SBSS approach. DUPLEX generates only two sets (train and test), and in most applications a 50:50 split between training and test data is assumed. *Snee* (1977) suggests that alternative splits could be achieved by allocating the remaining data to training when the test subset has been filled. In this study, a further modification was made to extend DUPLEX to the case of three sets, where pairwise sampling alternated between training, test and validating samples, until each of the smaller sets were filled. The steps of the algorithm are as follows:

i. Find $x_i, x_j \in \mathcal{D}$ that maximise the distance $\|x_i - x_j\|$ and sample (without replacement) into the training set, $\mathcal{T}$.

ii. Repeat once each for $\mathcal{S}$ and $\mathcal{V}$.

iii. Find next sampled pair x_i and x_j, such that they maximise the minimum single-linkage distance $\|x - s\|$ for each previously sampled $s \in \mathcal{T}$ and allocate to $\mathcal{T}$.

iv. Repeat, rotating allocation between sets $\mathcal{T}$, $\mathcal{S}$ and $\mathcal{V}$.

v. Once the two smaller sets are fully allocated, allocate remaining data to the largest set.

Here, in order to initialise each sample, the initialisation sequentially finds the pair of data points that lie farthest from each other within the database $\mathcal{D}$; the first pair allocated to the training data, and the second to the test data. Data are then sampled pair-wise, based on the maximum distance to the respective target set.

Stratified Systematic Sampling

Systematic stratified sampling has been used in a number of ANN development examples. The methodology is quite straightforward to implement and is implicitly a form of stratified sampling. In some cases, this may be easier or more convenient than the SBSS approach, and so was also considered for comparison. The algorithm proceeds as follows:

i. Sort the order of data in D by ascending y.

ii. Determine sampling interval $k = N/(N_{\mathcal{T}} + N_{\mathcal{S}})$.

iii. Randomly select start location $m \in [1, k]$.

iv. Draw every $m + k$ sample into $\mathcal{T}$.

v. Unsampled data are allocated to $\mathcal{V}$.

vi. Repeat steps 2 to 5 to sample $\mathcal{S}$ from $\mathcal{T}$.

Here, in order to generate the three samples, the systematic sampling is performed twice. First, training and test (calibration) data are sampled using the systematic approach, and then the test data are drawn from this sample.

Optimisation-based Data Splitting

Following the methodology of *Bowden et al.* (2002), an algorithm was implemented to minimise the dissimilarity of the data sets. It was observed that the specific GA implementation described by *Bowden et al.* (2002) behaved more as a random search of different combinations of data splits. In this study, the main aim was to test the assumption that the objective function optimises the sampling, by evaluating the bias/variance characteristics of the approach, and so a simple random search was adopted. In this approach, up to 10 000 independent data splits were randomly generated, with the optimal split found according the same objective function that was used by *Bowden et al.* (2002), which is to minimise the expression:

$$J_\mu + J_\sigma, \tag{8.17}$$

where J_μ and J_σ are given as

$$J_\mu = \sum_{i=1}^{d} |\mu_{i,\mathcal{T}} - \mu_{i,\mathcal{S}}| + |\mu_{i,\mathcal{T}} - \mu_{i,\mathcal{V}}| + |\mu_{i,\mathcal{S}} - \mu_{i,\mathcal{V}}|, \tag{8.18}$$

and

$$J_\sigma = \sum_{i=1}^{d} |\sigma_{i,\mathcal{T}} - \sigma_{i,\mathcal{S}}| + |\sigma_{i,\mathcal{T}} - \sigma_{i,\mathcal{V}}| + |\sigma_{i,\mathcal{S}} - \sigma_{\mathcal{V}}|. \tag{8.19}$$

Here, $\mu_{i,\mathcal{T}}$, $\mu_{i,\mathcal{S}}$ and $\mu_{i,\mathcal{V}}$ denote the mean of x_i in the training, testing, validating data sample, respectively; and similarly for standard deviation $\sigma_{i,\mathcal{T}}$, $\sigma_{i,\mathcal{S}}$ and $\sigma_{i,\mathcal{V}}$.

8.5 Results

Table 8.3 and Table 8.4 summarise the bias and variance of the GRNN error due to sampling for all of the sampling techniques for the case of uncorrelated input variables (Dataset I, II and III). The results show a clear difference between the

Table 8.3: Variability of ANN generalisation performance for split-sample validation using alternative sampling techniques. The regression task is the Friedman surface where input data are uncorrelated noise, and the sampling fraction is $(n/N) = 40\%$

	Dataset I		Dataset II		Dataset III	
Method	E	V	E	V	E	V
SRS	20.9	21.4	18.2	3.1	16.4	0.8
SBSS(S)†	27	8.7	23.5	4.9	20.6	1.3
SBSS(S)*	24.2	12.8	23.4	4.9	21.9	2.7
SBSS(SL)	25.5	16	19.6	0.2	19.4	1.6
SBSS(P)	20.8	15	18.7	2.4	17.4	0.9
SBSS(P)*	20.9	16.5	18.7	2.4	16.9	0.8
SBSS(N)	12.3	6.1	15.7	1.5	17	0.8
SBSS(N)*	13	14.6	15.7	1.5	15.8	0.6
DUPLEX	11.51	-	12.22	-	13.4	-
Systematic	20.2	7.5	19.7	0.2	17.7	0.2
Optimisation	21	18.1	22.6	5.4	16.4	0.6

† parentheses denote SBSS using different allocation rules: (S)=single, (SL)=single with large SOM, (P)=proportional, (N)=Neyman. ∗ denotes SBSS using a single instance of a SOM partitioning.

stratified sample allocation rules. DUPLEX and SBSS(N) produced the lowest model error in nearly all cases, demonstrating that this sampling technique consistently provided training, test and validation data that led to accurate GRNN models. In general, DUPLEX provided benchmark sampling performance, with a slightly lower model error than SBSS(N) and nil variance. The worst performing technique was SBSS(S), which consistently resulted in the largest bias and the largest variance. The high bias achieved using this sampling method is conclusive evidence that by drawing a single sample from each SOM unit, the data are under-sampled and a representative training sample is not obtained. In relative terms, the average bias of SBSS(S) was double that of SBSS(N), and this sampling technique also performed worse than the SRS benchmark. SBSS(SL), where the SOM grid was sized equal to the number of samples required and single allocation was used, resulted in poor sample quality. This demonstrates that taking a single sample from within the SOM map units does not yield a sufficiently representative set of training data, and could not be recommended in preference to Neyman, or even proportional sample allocation.

The results for systematic stratified sampling in Table 8.4 are characterised by a high bias, but low variability. The low variability can be attributed to the restricted number of samples that can be drawn using this non-probability sampling technique, which is determined by the number of possible start locations for

Table 8.4: Variability of ANN generalisation performance for split-sample validation using alternative sampling techniques. The regression task is the Friedman surface where input data are uncorrelated noise, and the sampling fraction is (n/N) = 80%

	Dataset I		Dataset II		Dataset III	
Method	E	V	E	V	E	V
SRS	18.6	66.4	16	5.6	14.1	1.5
SBSS(S)†	27	8.7	23.5	4.9	20.6	1.3
SBSS(S)*	24.2	12.8	22.7	5.4	21.9	2.7
SBSS(SL)	29.5	28	16.8	0.5	18.6	1.6
SBSS(P)	17.5	50.9	16.2	5.4	14.4	1
SBSS(P)*	18.8	68.2	16.2	5.9	14.3	0.8
SBSS(N)	6.3	0.1	10.2	0.3	15.4	0.6
SBSS(N)*	7.7	1.5	11.4	2.3	13.1	0.9
DUPLEX	27.88	-	18.79	-	16.75	-
Systematic	15.6	3.6	16.8	0.5	13.8	0.6
Optimisation	17.6	52.9	22.6	5.4	14.3	1.3

the sampling sequence, m. Although the expected error ranged from 13.8–16.8, which compared favourably to SRS (14.1–18.6), the minimum error achieved by systematic stratified sampling was 12.7–13.7, which was greater than that obtained using SRS (9.8–11.5). This result suggests that systematic sampling consistently resulted in a model with a relatively high error, where the average error of SRS was high more so due to the high variability of the sample, since high variability will also increase the expected error.

The optimisation algorithm resulted in a high bias and variance, and this result demonstrated that minimising the statistical difference of the datasets did not strictly ensure representative subsets. The lack of correlation between the global statistics and sample quality is evidence that the overall error is dependent on the accuracy of predictions within local regions. This is supported by the superior results obtained by methods such as SBSS and DUPLEX, which consider the local distribution of data and are therefore more effective at obtaining representative data.

The performance of the sampling methods on the correlated datasets (Datasets IV, V and VI) are shown in Table 8.5 and Table 8.6. The correlation effectively reduced the dimensionality of the input space and this had a significant influence on the relative performance of all sampling methods, as can be seen in the comparison between Table 8.4 and Table 8.6. For correlated data, the average error for all sampling methods was lower than the error for uncorrelated data. The

Table 8.5: Variability of ANN generalisation performance for split-sample validation using alternative sampling techniques. The regression task is the Friedman surface where input data are correlated noise, and the sampling fraction is (n/N) = 40%

	Dataset IV		Dataset V		Dataset VI	
Method	E	V	E	V	E	V
SRS	13	21.9	7.7	14.2	3.4	0.4
SBSS(S)	15.4	14.2	13.2	25.2	4.8	0.4
SBSS(P)	12.5	11.9	6.6	5.8	3.6	0.1
SBSS(N)	7.2	7.4	6.5	4.6	5	2.9
DUPLEX	3.9	-	3.15	-	2.51	-
Systematic	11.4	10.1	7.8	6.7	3.7	0
Optimisation	12.3	26.4	7.7	13.6	3.4	0.1

Table 8.6: Variability of ANN generalisation performance for split-sample validation using alternative sampling techniques. The regression task is the Friedman surface where input data are correlated noise, and the sampling fraction is (n/N) = 80%

	Dataset IV		Dataset V		Dataset VI	
Method	E	V	E	V	E	V
SRS	9.1	57.9	4.7	10.4	2.9	0.2
SBSS(S)	15.4	14.2	13.2	25.2	4.8	0.4
SBSS(P)	9.3	60	4.6	8.7	2.9	0.1
SBSS(N)	4.9	4.3	4.3	1.3	3.6	0.3
DUPLEX	4.8	-	4.9	-	3.1	-
Systematic	5.3	14	3.7	0.4	2.7	0.2
Optimisation	17.6	52.9	4.9	11.2	2.8	0.1

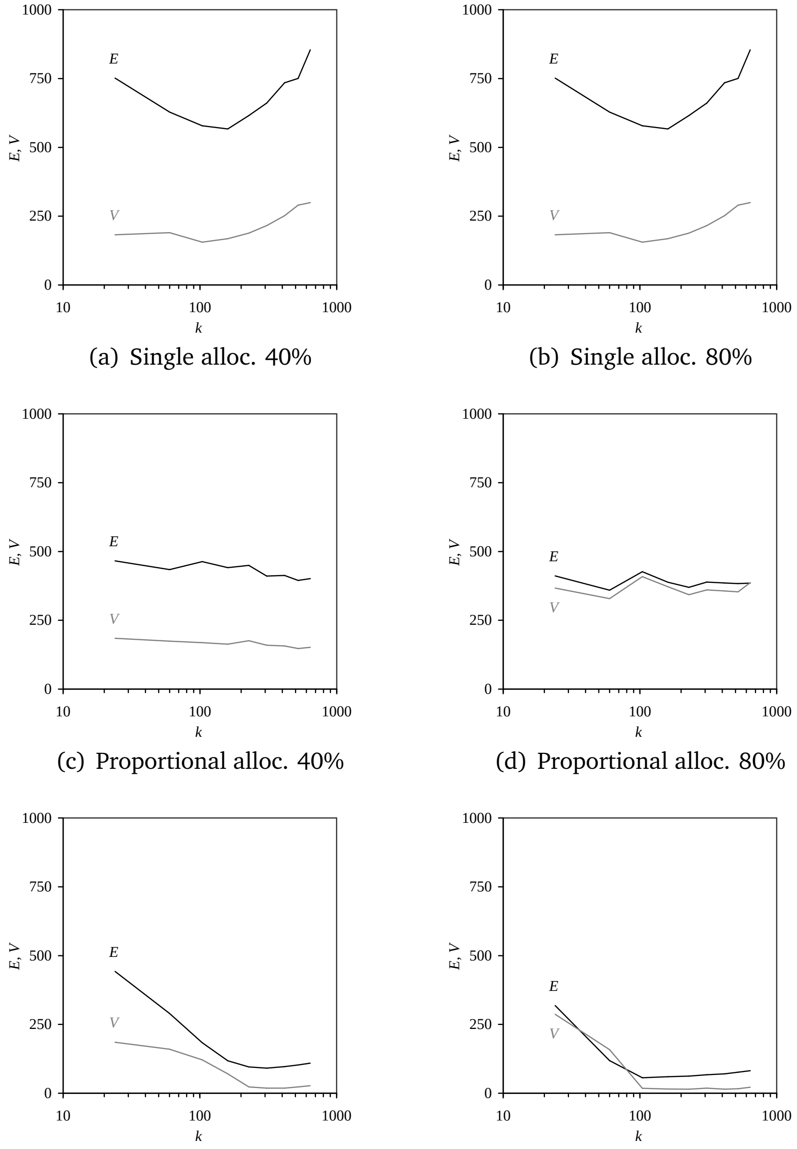
explanation for this is that the variability of sampling data is largely dependent on the variance of the joint distribution of the data. In the case of the correlated input data, the joint distribution has significantly less variance, since the input data have only one “true” dimension, corresponding to the independent input variable. The important consideration for sampling is therefore the dimensionality of the joint distribution of the data, as this predicts the difficulty of the sampling task.

The sensitivity of SBSS to the SOM grid size was investigated by estimating the sampling bias and variance for a SOM specified with a map size ranging from 20 to 640 units, with a ratio of approximately 1:6 set for the length of the sides. The results are summarised in Figure 8.2, which plots the bias and variance of model performance when SBSS data splitting was applied using each of the sample allocation rules, and for sampling 40% and 80% of the data into the calibration (training and test) datasets.

A clear optimum map size was observed for both SBSS(S) and SBSS(N), corresponding to minimum bias and variance, as shown in Figure 8.2(a), 8.2(b), 8.2(e) and 8.2(f). As the map size was increased above the optimum grid size, the bias and variance increased in the case of SBSS(S), but remained low for the case of SBSS(N). The bias and variance of SBSS(P) was relatively insensitive to the grid size, although there was a slight decrease in both bias and variance over the range of grid sizes tested. The optimum size SOM for SBSS(N) was approximately 220 map units for $n/N = 40\%$ (Figure 8.2(e)), and 100 map units for $n/N = 80\%$ (Figure 8.2(f)).

The values of the QE and silhouette coefficient for SOM sizes ranging from 2 to 640 units are shown in Figure 8.3. The silhouette coefficient, S, predicted a small number of clusters in each case, which has previously been suggested by *Kingston* (2006). The maximum value of S obtained was approximately 0.5 for a 2×1 SOM applied to Dataset I. In this case, the analysis of S correctly identified that Dataset I comprised two clusters, which correspond to the two Gaussian distributions that were sampled to generate the data. However, in most cases the value of S was approximately 0–0.1 for all SOM sizes larger than 3×1 units. According to the guidelines in Table 8.1, these values indicate that the clustering of the data is at best weak for such a large number of partitions.

The QE followed a typical decreasing trend for an increasing grid size, with an initially large relative decrease in QE, becoming smaller for successive increases in grid size. The minimum QE was observed for the largest SOM size of 640 units (32×20), although the trend indicated that the QE would continue to decrease for larger sizes, which was also not unexpected. Although the QE did not identify any clustering within the data, it is interesting to note that the observed trend was similar to the bias and variance trend over the same range of SOM sizes (Figure 8.2(e) and 8.2(f)). As can be inferred from the slope of the QE trend in Figure 8.3, the relative decrease in QE becomes small at approximately 100–200 map units for the datasets studied, which corresponds closely to the optimal number of map units determined based on bias and variance.



(a) Single alloc. 40%

(b) Single alloc. 80%

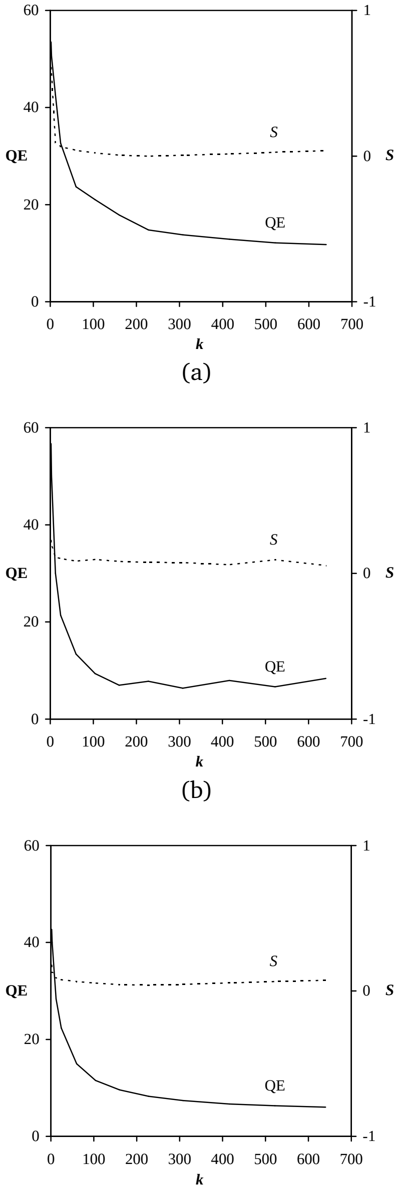
(c) Proportional alloc. 40%

(d) Proportional alloc. 80%

(e) Neyman alloc. 40%

(f) Neyman alloc. 80%

Figure 8.2: Effect of SOM size on validation error for data sampled using SBSS with single, proportional and Neyman allocation.



(a)

(b)

(c)

Figure 8.3: Silhouette coefficient (S) and quantisation error (QE) versus k for SOM partitioning of (a) Dataset I, (b) Dataset II, and (c) Dataset III.

8.6 Discussion

8.6.1 Factors influencing data splitting performance

The study presented in this paper has not only presented a thorough evaluation of the SOM-based approach for data splitting, but has also compared its performance against other approaches. Furthermore, the study has also compared how the performance of different data splitting methods are influenced by the proportion of data sampled, and the dimensionality and distribution of the data.

Comparisons of results in Table 8.3 and Table 8.4 show that, with the notable exception of DUPLEX and SBSS(S), model error was improved by sampling 80% into training and test data. Overall the results indicate that maximising the amount of data used for model development improves the accuracy of the model, which is to be expected. However, SBSS(N) and DUPLEX provided a significantly lower bias and variance than the other sampling techniques for the lower sampling fraction of 40%, indicating that these methods were more efficient at selecting representative data. This is an important result, since in many applications there is a need to minimise the number of data samples used to maintain reasonable computation times. SBSS(N) and DUPLEX therefore provide useful sampling techniques that can ensure that the quality of the hold-out validation is less affected by sampling fewer data.

The relative difference in the performance of the GRNN was significantly influenced by the distribution of the data. In particular, SBSS(N) was the best sampling method for Dataset I, in which the distribution of the input data was highly skewed. This high quality of sampling for Dataset I can be attributed to the ability of the Neyman allocation rule to over-sample sparse data. However, for Gaussian data (Dataset II) the sampling methods were more closely matched, and for uniform data, no sampling method was found to provide a significant improvement over SRS. This result indicates that when there is no structure within the data, there is not likely to be any benefit in performing stratification of the input data. An interesting result was the decrease in performance of DUPLEX when 80% of the data were sampled, as this method gave the best results in all other cases. This result was unexpected, although it can be explained by the insufficient sampling of training data in sparse regions, which resulted in poor test and validation results within these regions.

The relative performance of different sampling techniques was also affected by the number of dimensions of the sampled data. In particular, the relative gain of SBSS(N) over systematic sampling is significantly reduced for the case of correlated input data. DUPLEX and SBSS(N) produced the lowest bias and variance for Dataset IV, confirming the benefit of these sampling techniques when applied to skewed distributions. However, for datasets with Gaussian (Dataset V) and uniform (Dataset VI) distributions, systematic stratified sampling gave good results and had lower bias and variance than SBSS(N). It can be concluded that in low (single) dimensional problems, systematic stratified sampling is able to provide a more consistent and more representative sample of the data, since samples are distributed evenly over the data space, which is relatively easy to achieve in this manner for one dimensional, smoothly distributed data.

8.6.2 Selecting a suitable data splitting approach

Several conclusions were made based on the comparisons between the sampling techniques investigated, that have been formalised as a set of guidelines for choosing an appropriate sampling technique for the generation of training, test and validation data for ANN development. The guidelines, which are summarised in Figure 8.4, are proposed so that the sampling technique applied will result in minimum bias and variance of the hold-out validation method, and to increase confidence in the results obtained and the model developed. Applications with non-uniformly distributed data will benefit from sampling techniques that can adjust the sampling rate throughout the data subspace to ensure adequate representation of all conditions. The choice of sampling technique may be less important for data that are uniform, although this rarely is the case.

SRS is not recommended as a suitable sampling technique for the generation of training, test and validation data samples. In particular, the inability of SRS to draw a reliable sample for skewed or non-uniformly distributed data results in poor hold-out validation performance. The method is also highly variable, which reduces confidence in the results of one-off testing and validation that are obtained during ANN model development.

For the sampling of highly multivariate, non-Gaussian and non-uniform data, SBSS will yield good results, since the stratification considers all dimensions and can therefore provide a sampling frame that provides representation of all features. In the case that there may be non-uniformly distributed, or non-Gaussian data, Neyman allocation will yield a sample with the minimum variance. The Neyman allocation rule is able to account for variations in density in the data to ensure sparser regions are over-sampled. However, in the case of multivariate Gaussian data, proportional allocation will yield similar results. The DUPLEX algorithm provides a good approach to data splitting when data are Gaussian or

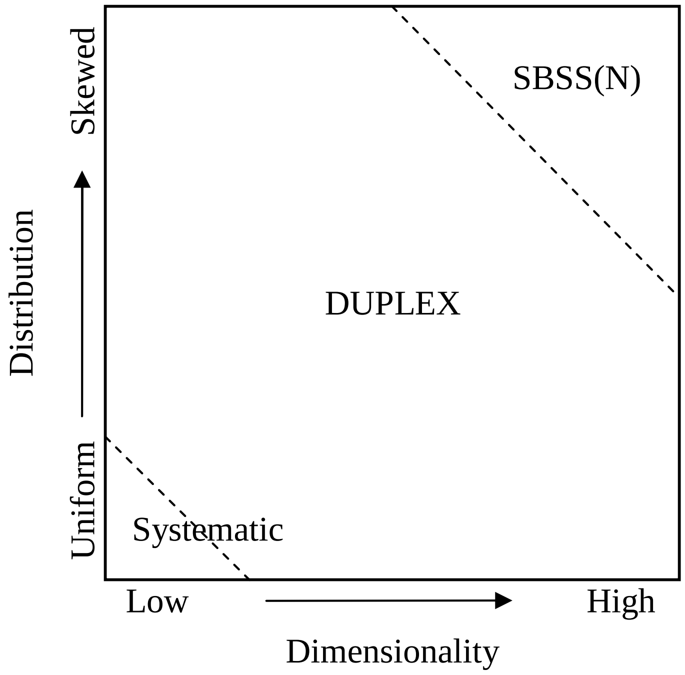


Figure 8.4: Guidelines for choosing an appropriate sampling technique based on the dimensionality of the data, and the distribution of the variables.

uniformly distributed, and although the expected cross-validation performance is only slightly better than that of SBSS(N), it does not suffer from variance, which will increase confidence in the results obtained using this approach. However, although the computation associated with DUPLEX was not prohibitive for the datasets used in this study, for large datasets the SBSS(N) approach is likely to yield a more efficient data splitting methodology.

The benefit of SBSS may not be as significant in low dimensions, and so correlations between input dimensions need to be considered. The effect of correlations effectively reduces the dimensionality of the problem, since there is overlap of the distributions of one or more variables. In this case, DUPLEX appears to offer best results, as it can effectively distribute samples throughout the database. Systematic stratified sampling is also an efficient method for selecting data for ANN models in the case of low-dimensional data, that is data with only one input variable, or one dominant input variable, since systematic stratified sampling stratifies the data on a single variable. Although the method is non-probabilistic and therefore restricts the number of possible samples that can be obtained, it was found to result in low bias and variability in these cases, and therefore is able to consistently generate a good sample. In terms of computational efficiency and simplicity, the method is much easier and faster to implement than the more complex DUPLEX and SBSS(N) algorithms. However, as dimensionality increases, it should be noted that systematic sampling may not provide a sampling frame that captures all features in multivariate space, and would be recommended only for univariate or bivariate datasets.

8.6.3 Specification of SOM Parameters

A shortcoming of previous work has been the lack of guidelines for determining how to specify the SOM in order to achieve good data splitting. The two main issues being that of specifying the SOM size and how to allocate the data within SOM units to the training, test and validation samples. The results of this study indicate that the bias and variance of model performance is significantly influenced by the size of the SOM. The results suggest a more moderate grid size might be favoured, but is entirely dependent on the number and distribution of available data, and on how many data are to be drawn. The dependence of the optimal number of SOM map units on the sampling fraction n/N may be explained by the sensitivity of the sample allocation rules to the number of data within each stratum. If the sampling quota is higher in each stratum, then potentially the same representativeness can be gained with slightly fewer and larger strata due to the drawing of a larger sample. In order to obtain an equally representative sample using a smaller quota, a finer sampling frame (i.e. using a larger number of map units) is required to ensure data are sampled adequately throughout the domain, in order to reduce the effects of random sampling within each stratum.

It is evident from this study that the application of traditional CVIs to clustering data for implementing sampling techniques for ANN training data selection is likely to be somewhat limited by the degree of clustering within the available data. Although in this study, only the silhouette coefficient was applied, a similar result would be expected for all CVIs, as they similarly estimate the relative compactness and separation of clusters to identify the natural number of clusters. In the traditional sense, the number of natural clusters within a set of regression data may be low. However, the goal of the partitioning in CBSS is to define regions that are sufficiently homogeneous to optimise the data sampling, and it has already been demonstrated that a very small number of partitions does not yield a good sample.

The correspondence of the map size determined using the heuristic grid size formula in (8.11) with the optimal bias and variance is also reasonably close. The heuristic rule predicts 208 map units for SOM partitioning of 1 000 samples, with $\beta = 5$. This predicted number of units is close to the optimal number observed for

SBSS(N) with 40% allocation, although it was twice the optimal size of 100 map units that was obtained for 80% allocation. Based on the results in this study, it appears that the degree of partitioning required is greater when fewer samples are drawn, and specifying too few map units would appear to have more severe consequences than too many. The result is consistent with the notion of ensuring adequate coverage of the data. The heuristic rule may therefore provide a useful means of specifying the size of the SOM without undertaking extensive trial and error.

8.6.4 Effect of SOM initialisation

Based on the results obtained for SBSS(S)*, SBSS(P)* and SBSS(N)* in Table 8.3 and Table 8.4, it was concluded that the contribution to sample variability due the random initialisation of the SOM was relatively small, since there was little difference between the cases where independent SOM partitionings were used, and the cases where the partitioning was fixed. If anything, the results show higher bias and variance for the case where the SOM partitioning was fixed, which is quite unexpected. The result may be attributed to the quality of the partitioning achieved in this instance in some way resulting in a poorer than average sample. However, overall the results do not indicate that the SOM was a significant source of variation, but rather that the SOM partitioning is a reasonably stable and reliable algorithm for performing the stratification. However, it should be noted that any improvement in the SOM algorithm, in terms of yielding a consistently high quality sample, would be worthwhile. Alternative initialisation schemes may reduce the variability, and detailed investigation of learning algorithm behaviour (i.e. effects of the neighbourhood and map edge effects) could also yield further improvements in the SBSS approach.

8.7 Conclusions

Hold-out validation is the most common method used to ensure generalisation is achieved during ANN model development. The importance of good sampling is apparent, when considering data splitting typically involves a single sampling of the available data to form the subsets, which subsequently underpins ANN training and validation. The implications of the choice of sampling methodology used for data splitting are becoming increasingly recognised, and improved sampling techniques are being sought to overcome potential bias and variance that arise from sampling data that are non-uniformly distributed.

In comparison to other sampling techniques, a multivariate stratified sampling approach based on the SOM appears to be suitably robust and consistently produce superior ANN models. However, the efficiency of SBSS was only evident in multivariate, non-uniform datasets. DUPLEX otherwise provides a benchmark for data splitting, and generates representative datasets and low model bias. Furthermore, the approach is deterministic and so there is no variance in model performance. Alternatively, a simple and fast systematic stratified approach, by sorting data along the predicted variable, was found to yield good performance for low-dimensional datasets. In either case, these sampling techniques can draw a representative sample even for skewed distributions of data and provide significant improvements over simple random sampling.

In reviewing the implementation of SOM-based stratified sampling, the issues of sample allocation and map size were examined with some interesting results. The SOM quantisation error (QE), rather than traditional cluster validity measures, appears to be more useful in determining the required map size to generate a high quality sample. A rule-of-thumb formula for the map size was also found to provide a reasonable estimation of the number of map units that corresponded to minimal bias and variance, which can be used to specify the grid size without the need for extensive trial and error. Neyman allocation, which increases the sample quota for sparse map units, generated significantly more reliable sampling than other techniques. Since the SOM learns to approximate the distribution of the data, the SBSS technique with Neyman sample allocation provides a way to implement density biased sampling without having to determine the distribution of the data. Although not explored in this paper, it is quite conceivable that the sample allocation could be further modified to allow for tuning for the requirements of a specific application, resulting in a generalised sampling approach.

Acknowledgements

This research was jointly supported by United Water International Pty Ltd and the Cooperative Research Centre for Water Quality and Treatment.

Chapter 9

SOMPLEX: A hybrid SOM-DUPLEX data splitting algorithm for ANN development

Publication 5

Publication Details

This work was submitted to the journal *Water Resources Research* on September 28, 2009, as the following article:

> May, R. J., H. R. Maier, and G. C. Dandy, SOMPLEX: a hybrid SOM-DUPLEX data splitting algorithm for ANN development, *Submitted to Water Resources Research*.

Although the manuscript has been reformatted in accordance University guidelines, and sections have been renumbered for inclusion within this thesis, the paper is otherwise presented herein as submitted.

Statement of Authorship

May, R. J. (Candidate)
Literature review, conceptual development, experimental design, analysis and interpretation of results, preparation of manuscript and corresponding author.

Signed: .... Date: ....

Maier, H. R.
Research supervision and review of manuscript.

Signed: .... Date: ....

Dandy, G. C.
Research supervision and review of manuscript.

Signed: .... Date: ....

Abstract

In applications of ANN models in water resources modelling, hold-out or early stopping *remains the most common technique to ensure generalisation from ANN models constructed on a limited sample of historical data. During model development, data splitting is employed to generate independent training, test and validation data. However, depending on the data splitting algorithm, this procedure can potentially introduce sampling bias and variance, which undermines confidence in model performance.*

In this paper, a novel multi-stage data splitting approach called SOMPLEX is proposed that combines clustering the self-organizing map (SOM) with DUPLEX sampling. The SOM has recently been explored as a tool for data splitting, as it can define partitions within a database to support stratified sampling. DUPLEX is a data splitting method that provides uniform coverage over a sampled database. Results for a comparative study are given that benchmark the performance of SOMPLEX against several popular data splitting approaches for a number of real-world ANN water resources modelling tasks. Relative to each method alone, the SOMPLEX approach achieves highly reliable ANN generalisation performance, while maintaining excellent computational scalability and flexibility. The SOM partitioning reduces the computational complexity of DUPLEX, which in turn provides a deterministic approach to splitting data within the SOM partitions.

9.1 Introduction

Artificial neural network (ANN) models have become a popular approach to the modelling and analysis of complex, non-linear environmental systems. In the field of water resources, ANN models have been used widely in applications such as water supply management (*Adeloye*, 2009), rainfall-runoff modelling (*Hsu et al.*, 2002; *Jain et al.*, 2004; *Lauzon et al.*, 2006; *Jain and Srinivasulu*, 2006), flood forecasting (*Laio et al.*, 2003; *Dawson et al.*, 2006), streamflow forecasting (*Coulibaly et al.*, 2000; *Hu et al.*, 2001; *Sivakumar et al.*, 2002; *Wang et al.*, 2006), water and wastewater treatment process modelling (*Machon et al.*, 2007; *Raduly et al.*, 2007; *Maier et al.*, 2004; *Baxter et al.*, 2000), waste water collection system management (*Darsono and Labadie*, 2007), and water quality prediction (*Ani et al.*, 2009; *May et al.*, 2008b; *Alp and Cigizoglu*, 2007; *Maier and Dandy*, 1996; *Serodes et al.*, 2001; *Bowden et al.*, 2005).

A key issue during the development of ANN models is to ensure good generalisation. Generalisation refers to the ability of any statistical model, whose parameters are estimated on a limited set of training data, to accurately predict on novel or previously unseen data. Because ANN models can possess a large number of parameters, relative to the number of available training data, they are particularly susceptible to the problem of over-fitting, which leads to optimistic estimates of accuracy that is characterised by poor validation performance (*Sarle*, 1997). Various cross-validation techniques can be used to ensure that ANN models do not over-fit the training data (*Anctil and Lauzon*, 2004). By far, the most common methodology employed is the train-and-test cross-validation approach, or *early-stopping*. In this approach independent sets of data are used during training and testing, and training occurs until the minimum test error is determined, which infers the best degree of generalisation. Model selection, which is often necessary to determine the optimal ANN architecture, can be optimistically biased towards the test data used to evaluate and compare model performance. The selected ANN model must therefore be validated on a third independent set of data to obtain an unbiased validation of model performance.

In order to implement cross-validation, data splitting is commonly employed to generate pseudo-independent data sets for training, testing and validating the model, from a single database. However, the data splitting method employed can have a significant impact on the training, testing and validation of ANN performance. The data splitting must ensure that data allocated to each set are equally representative of the modelling domain. Failure to do so leads to a *bias* in the training, testing or validation performance assessment. Furthermore, the data splitting must consistently generate representative datasets. Inconsistent data splitting produces *variance* in the generalisation performance, which creates uncertainty regarding model performance. These issues are rarely considered in many ANN applications, but given that data splitting generally occurs only once in ANN development, it is highly important to determine appropriate methods for data splitting that reduce the potential for variation in model performance, and reduce the bias of test and validation performance.

Various algorithms have been described for performing data splitting during ANN development, but relatively few studies have considered the performance of these approaches in terms of bias and variance. Consequently, this paper first reviews the different data splitting approaches and critically evaluates their relative benefits and shortcomings within the context of ANN development. Based on this review, a newly proposed algorithm called SOMPLEX is described that improves upon two existing data splitting approaches, drawing from their individual strengths. A comparative study is described that evaluates the relative performance of the new and existing approaches for some real-world examples in the field of water resources modelling and analysis.

9.2 Data Splitting Methods

Past reviews have considered some of the approaches used to perform data splitting within ANN applications, and found that the approaches could be considered as: random, judgemental (heuristic), or trial-and error (*Bowden et al.*, 2002; *Shahin et al.*, 2004). A taxonomy of data splitting approaches that have been applied to ANN development is shown in Figure 9.1. This taxonomy follows a similar classification to sampling algorithms, since data splitting is essentially a sampling application. The three main classifications are: random (probability), deterministic (non-probability), and multi-stage. Each of the various approaches have their benefits and weaknesses, which are discussed in the following subsections. The main considerations are the bias and variance of the sampling, which reflects the ability of the data splitting approach to reliably split the data into equally representative sets. However, other issues to be considered are the amount of required knowledge regarding the data, algorithm complexity, suitability for different types of datasets and applications, and computational effort.

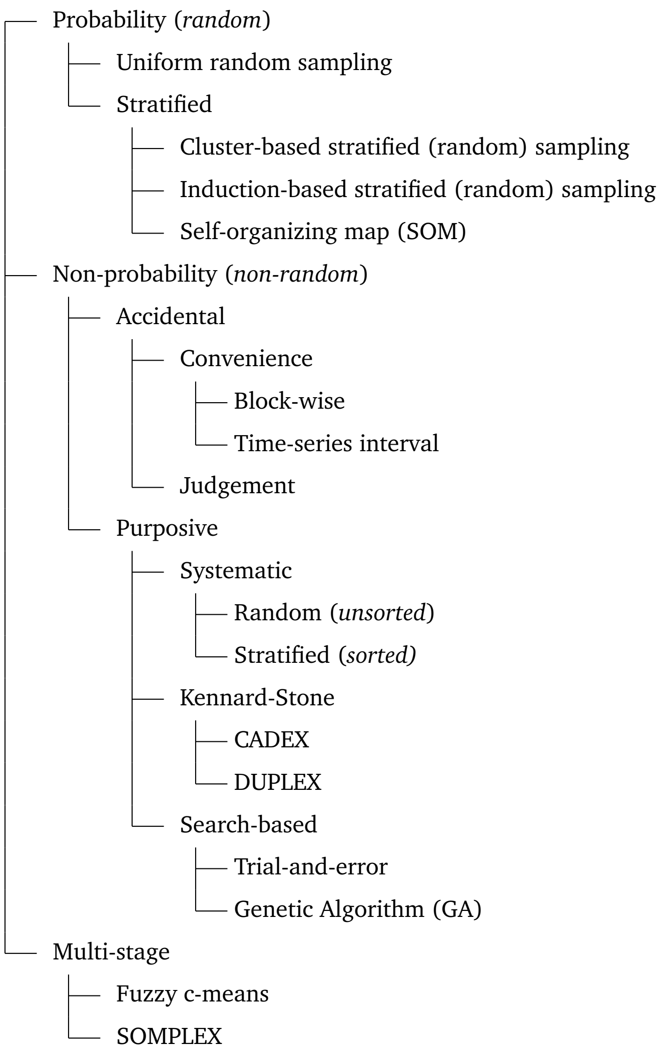


Figure 9.1: Taxonomy of data splitting algorithms used in ANN development.

9.2.1 Uniform random

Uniform random sampling is the most common method for implementing data splitting across many statistical applications, including ANN development. In this approach, data are sampled with uniform probability (without replacement) into training, testing and validating data sets. The characteristics of the data selected are not considered, and so a good or poor split may result purely by chance. In particular, random sampling performs poorly when the data distribution is skewed, and the selected data can be heavily biased towards more frequent cases within the database (*Kollios et al.*, 2003). This bias in the training set results in poor predictions for rare cases, that may be overlooked by the random sampling (*Tourassi and Floyd*, 1997). However, ensuring accurate predictions of low-frequency events is often an important issue when considering the application of ANN models to the prediction of rare or extreme events, such as failure analysis, or flood forecasting.

The potential for a poor split and a highly variable sample is a particular concern for ANN development, since the data splitting is performed once during development (*Verstraeten and Ven den Poel*, 2006). Any doubt surrounding the representativeness of a given data split undermines confidence in the latter stages of model development and assessment of model performance.

9.2.2 Stratified

Stratified sampling is a methodology that is often applied to improve the quality of random sampling for statistical data analysis (*Cochran*, 1977). The stratification identifies distinct, but homogeneous regions within the database, and forces random sampling of data from within each stratum. Furthermore, by adjusting the number of data drawn from different strata, the sampling can be tuned to yield an optimally representative sample (*Cochran*, 1977). For multivariate datasets, clustering algorithms provide a convenient way to stratify the available data (*Mulvey*, 1983). Several approaches have been considered for ANN data splitting, with variations on the number of clustered variables and the clustering algorithm used. For example, induction-based stratified sampling stratifies the data on the single most relevant input variable (*Gill et al.*, 2004). Clustering algorithms that have been used include k-means, DBSCAN (*Daszykowski et al.*, 2002) and fuzzy c-means clustering (*Shahin et al.*, 2004).

The self-organizing map (SOM) (*Kohonen*, 1995) has been proposed as a tool for data splitting, with several recent examples in ANN modelling applications (*Wong*, 1996; *Daszykowski et al.*, 2002; *Bowden et al.*, 2002; *Anctil and Lauzon*, 2004; *Zhang et al.*, 2004b). Related work also includes some examples of SOM clustering, where separate ANN models are subsequently developed using data from each cluster (*Hsu et al.*, 2002; *Jain and Srinivasulu*, 2006).

The SOM is an unsupervised ANN that is used widely in a number of data analysis applications (*Kalteh et al.*, 2008). The SOM is an array, or *map* of weights, or *codebook vectors*, corresponding to p-dimensional space. The SOM is trained to learn the positioning of its codebook vectors that best describes the distribution of a p-dimensional set of training data. Data are clustered onto the SOM according to their nearest codebook vector, which are then projected onto the 2-dimensional map. In data splitting applications of the SOM, the data are first clustered on the SOM, and then data are selected from each map unit to ensure that training, testing and validating samples are representative. In the taxonomy in Figure 9.1, the SOM data splitting approach is a form of stratified sampling.

SOM-based stratified sampling (SBSS) appears to be an attractive option for data splitting, since the SOM is a clustering approach that ANN practitioners may already be familiar with, and will be readily able to implement. However, few of the reported applications of the SOM provide details on the specific implementation, and the use of the SOM still presents a challenge. Firstly, the sampling is sensitive to the SOM algorithm parameters and map size, and the partitioning that results. However, there are no guidelines for choosing optimal SOM parameters for data splitting (*Bowden et al.*, 2002). Secondly, the best approach to selecting from within the SOM map is not clear (*Daszykowski et al.*, 2002). In each of the approaches reported within the literature, different methods were used for specifying the SOM, and the data were selected differently. In addition, several examples neglect to fully describe the details of the implementation of the SOM data splitting approach used. *May et al.* (2009b) provides details on methods to select the size of the SOM, and how to determine the number of samples to draw from each SOM partition in order to minimise sample bias and variance for random stratified sampling of the SOM. However, stratified random sampling of the SOM is inherently probabilistic, and although it is reduced in comparison to uniform random sampling, a small amount of variation in performance can still be observed.

9.2.3 Convenience

Many ANN practitioners also utilise data splitting methods that are classified in statistical terminology as *convenience* sampling. Block-wise sampling refers to the splitting data by simply partitioning the data arbitrarily into training, test and validating samples of prescribed proportions. While convenience sampling methods are easily repeatable and tend be characterised by low variance, they can potentially be highly biased. A common example of convenience sampling occurs in time-series analysis, when contiguous time intervals are selected for training, testing and validating data (*Imrie et al.*, 2000; *Coulibaly et al.*, 2000; *Hu et al.*, 2001; *Jain and Srinivasulu*, 2006; *Cigizoglu and Kisi*, 2006; *Alp and Cigizoglu*, 2007). The concern with using such an approach is that there may be unique features that occur within each time period. *Flood and Kartam* (1994) showed that using training data that are biased towards a particular season, which did not fully represent all extremes of system behaviour, can result in a poor model. A study by *Bowden et al.* (2002) similarly illustrated the detrimental effect of using time-intervals to split data for ANN development, where data in the training data were found to be uncharacteristic of the validation data, which resulted in poor model accuracy. Likewise, convenience sampling of spatial data may suffer equally from this potential bias.

9.2.4 Judgement

Some methods have relied on expert judgement or heuristic rules to decide how to partition the data (*Bowden et al.*, 2002). This approach requires extensive knowledge or familiarity with the system under consideration, which is not typically the case in data-driven modelling applications. Judgement is often biased by the modeller's experience and understanding of the system under consideration. Variation in performance can often result as the derived data splitting rules tend to be highly case-specific, or subjective, so that these approaches are generally not easily transferable across different applications.

9.2.5 Systematic

Systematic sampling is a deterministic sampling approach that draws every k^{th} datum for a given set. The number of possible ways to split the data using this approach is limited to k, depending on the location of the first sample drawn. If the data are unordered the data split may be random. However, the risk with performing systematic sampling is that the sampling interval coincides with some natural periodicity in the data, in which case the resulting data sets will be heavily biased towards a particular part of the seasonal cycle. This is particularly a risk in time-series analysis where data are often found to be highly seasonal.

Baxter et al. (2001) describe an approach to data splitting using systematic sampling, where the data are first sorted along the output variable. This effectively results in a sample that is implicitly stratified over the output variable range. However, while this provides a simple approach for implementing stratified sampling, a potential drawback is that the input-output mapping is assumed to be 1:1. In multivariate data analysis, this approach may not identify significantly different regions of the input domain that yield the same output.

9.2.6 Kennard-Stone

Kennard and Stone (1969) developed the CADEX and DUPLEX data splitting algorithms for split-sample validation of regression. These approaches are sometimes collectively referred to as *Kennard-Stone* data splitting. CADEX proceeds by initially sampling the point that lies farthest from all others within the database into the calibration set, and the next farthest into the test set. Subsequent data are then iteratively sampled one-by-one, by identifying the datum that lies farthest from any previously selected points into the target set, where the target set alternates at each iteration between calibration and test set. DUPLEX is a modified form of CADEX in which data are selected in a pair-wise manner, which can reduce the optimism of the test data (*Snee*, 1977).

The Kennard-Stone sampling approach is fully deterministic and only one split is possible for any given database, resulting in zero sample variance. Conventional Kennard-Stone sampling generates a 50:50 split into two datasets, however it is possible to generate data sets of arbitrary proportions by allocating data to the smaller set until it is filled, and allocating all remaining data to the larger set (*Snee*, 1977). It is also possible to generate three sets, by rotating between three sets, rather than alternating between two.

Both CADEX and DUPLEX algorithms have been used widely in the field of chemometrics, including several applications to ANN development (*Despagne and Massart*, 1998). However, in comparison to other approaches, they are relatively unknown within the field of water resources or environmental modelling and analysis. The main limitation of Kennard-Stone sampling is the computational requirement of the algorithm. The approach has operational complexity O$\{N^3\}$, and memory complexity O$\{N^2\}$. Scalability to large datasets is therefore poor and may prohibit its use on many environmental datasets.

9.2.7 Search-based

Trial-and-error approaches have been used to perform data splitting, such that the training, testing and validating data are statistically similar, although the approach taken in order to achieve this is not always transparent (*Bowden et al.*, 2002). A data splitting technique based on genetic algorithms (GAs) provides an automated approach to determine optimally similar sets (*Bowden et al.*, 2002). The key to any of these approaches is to find a suitable definition of an *optimum* split. In general, optimality has been assessed in terms of statistical similarity. However, the notion that similar statistics lead to reliable, representative coverage of the multivariate input-output space has not been validated.

9.2.8 Multi-stage

Multi-stage sampling describes any approach that combines one or more sampling techniques to improve on a single technique. The most common basis for multi-stage sampling is the application of stratified sampling with deterministic sampling of data from each stratum, rather than random sampling. The aim is to achieve adequate global coverage of the data set that is achieved by stratification, but with improved sampling in local regions covered by each of the strata. This is beneficial if the strata are large, or if there remains some structure or diversity within the strata that needs to be adequately sampled.

Shahin et al. (2004) proposed a multi-stage data splitting approach that combines fuzzy c-means clustering with a heuristic rule for sampling data based on their fuzzy membership. The heuristic rule for selecting samples enables the number of samples drawn from each cluster to be adjusted depending on the spread of data, and forces the selection of data within each membership band to improve the coverage of the sample drawn.

The multi-stage approach provides the basis for an approach that combines clustering of the self-organizing map (SOM) with Kennard-Stone sampling. This approach, hereafter called SOMPLEX, is described in detail in the following section.

9.3 The SOMPLEX Algorithm

A multi-stage algorithm called SOMPLEX is proposed in this study, which combines clustering on the SOM with DUPLEX sampling of map units. SOM clustering is useful for identifying distinct regions of the modelling database for either particular classes or examples of unique input-output cases. The weakness of stratified random sampling is that data are assumed to be homogeneously distributed within each of the map units, and therefore random sampling within each cluster is adequate. However, due to the nature of the SOM, the distribution of data within the map units is rarely uniform. This is often overlooked, because the SOM is generally visualised by the topographical projection that only describes the number of data within each map unit, and not the distribution of data. Consequently, the interpretation of SOM mapping in many data splitting applications neglects the fact that map units may cover varying proportions of the database, and it is often the case that data in some partitions are more widely spread than in others, or that the distribution of data within a partition may be non-uniform (*May et al.*, 2009b). Given any distribution of data, DUPLEX can generate a uniform sample that provides total coverage of the data (*Snee*, 1977). In the case of data partitioned by the SOM, a representative sample can be drawn from each partition regardless of the heterogeneity or spread. The intra-cluster sampling is therefore robust and relatively insensitive to the partitioning that results from the SOM algorithm. Furthermore, since the DUPLEX algorithm is fully deterministic, there is no sample variance due to the intra-cluster sampling, which further improves on the conventional SOM approach.

An additional advantage of SOMPLEX is the scalability of the algorithm to large datasets, since the number of clusters remains proportional to $\sqrt{N}$. As mentioned in Section 9.2.6, the DUPLEX algorithm has an operational complexity $\sim O\{N^3\}$, and memory complexity $O\{N^2\}$, which causes the computational requirements of DUPLEX to increase very quickly for even modestly sized datasets. However, by using a SOM with $\sqrt{N}$ map units, the DUPLEX sampling considers clusters with an average size $\sqrt{N}$ so that the operational complexity for SOMPLEX becomes $\sim O\{N^{1.5}\}$, and the memory requirement is reduced to $O\{N\}$. Consequently, the SOMPLEX algorithm provides a highly efficient means of applying the DUPLEX approach to large datasets, which may present some use in ANN applications such as hydrological modelling.

Given some database $\mathcal{D}$, the SOMPLEX data splitting algorithm generates sets for training ($\mathcal{T}$), testing ($\mathcal{S}$) and validating ($\mathcal{V}$) as follows:

SOM clustering:

i. Initialise a $r \times c$ SOM

ii. Train the SOM on database $\mathcal{D}$

iii. Cluster $\mathcal{D}$ onto the trained SOM

iv. For each SOM unit $C^{(m)}$,

DUPLEX:

v. Set sampling quota

$$n_{\mathcal{T}}^{(m)} = \left|C^{(m)}\right| \frac{n_{\mathcal{T}}}{N}, \tag{9.1}$$

and similar for $n_{\mathcal{S}}^{(m)}$ and $n_{\mathcal{V}}^{(m)}$

vi. Initialise empty cluster samples $\mathcal{T}^{(m)}$, $\mathcal{S}^{(m)}$ and $\mathcal{V}^{(m)} = \emptyset$

vii. Find pair $x_i, x_j \in C^{(m)}$ that maximise $\|x_i - x_j\|$, and sample without replacement into $\mathcal{T}^{(m)}$

viii. Repeat step 7 for $\mathcal{S}^{(m)}$ and $\mathcal{V}^{(m)}$

ix. Find the next pair x_i and x_j, that maximise $\|x - s\|$, for $s \in \mathcal{T}^{(m)}$

x. Repeat step 9, rotating between $\mathcal{T}^{(m)}$, $\mathcal{S}^{(m)}$, and $\mathcal{V}^{(m)}$, until the quota for each set is filled.

xi. Merge $\mathcal{T}^{(m)}$ with $\mathcal{T}$, $\mathcal{S}^{(m)}$ with $\mathcal{S}$, and $\mathcal{V}^{(m)}$ with $\mathcal{V}$

Table 9.1: Specifications of the SOM

Parameter	Ordering	Tuning
Initial learning rate	0.9	0.01
Initial neighbourhood size	r	1
Epochs	2	20
Neighbourhood function	Gaussian	
Decay function	Linear	
Conscience	10	
Bias	0.0001	

Here, $n_{\mathcal{T}}$, $n_{\mathcal{S}}$ and $n_{\mathcal{V}}$ denote the specified number of data for sets $\mathcal{T}$, $\mathcal{S}$ and $\mathcal{V}$, respectively; which are determined by the specified proportions for each set and the number of data, N, in database $\mathcal{D}$. A proportionate number of data are selected from within each cluster, depending on their size $\left|C^{(m)}\right|$.

The SOM is specified with k map units in a rectangular grid of r rows and c columns. A heuristic rule $k = 2\sqrt{N}$ is used to determine the number of map units (*Vesanto and Alhoniemi*, 2000). The importance of each of the inputs, with respect to the output variable, can be used to determine the shape of the map, where the map length-to-width ratio should be similar to the ratio of importance for the two most important variables (*Cereghino and Park*, 2009). *Cereghino and Park* (2009) utilise the ratio of eigenvector lengths from principal component analysis (PCA) to determine the relative importance of variables.

The SOM is trained using the conventional SOM learning algorithm, which involves a short ordering phase with a large weight adjustment, followed by a longer tuning phase of fine adjustments. The specified SOM learning parameters used in this study are summarised in Table 9.1. Full details of the SOM algorithm are provided in *Kohonen* (1995). An important aspect is the inclusion of the conscience mechanism, in order to avoid the generation of large clusters and to ensure that the distribution of data is as even as possible.

9.4 Methodology

In this paper, the utility of SOMPLEX is evaluated by comparing its performance on five real-world datasets with that of several of the existing data splitting approaches described in Section 9.2. Past comparisons of data splitting approaches have been limited, and have focussed on similarity of the statistics of datasets generated, in order to define the ability of each approach to draw representative data (*Bowden et al.*, 2002; *Shahin et al.*, 2004). However, less attention has been paid to quantifying or comparing the performance of data splitting with respect to other issues, such as bias and variance.

Analysis of the bias and variance of data splitting algorithms provides a more rigorous evaluation of data splitting than examining the statistics of datasets obtained from a single split. Importantly, the sample variance provides an indication of the variability of model performance for independent resampling. A highly variable data splitting approach results in poor confidence for a one-off split during ANN development, as the performance obtained for a particular split could lie anywhere on a wide distribution of performance, which is often not known *a priori*.

The average model error relates to both bias and variance, and a high variance can contribute to a high average error. Examination of both average error and variance will indicate whether a model performance resulting from a particular data splitting method is consistently biased, or is simply highly unpredictable. A high error, combined with low variance, indicates poor representation of data within each of the data sets. In this case, the estimate of model performance is considered to be *pessimistic*. On the other hand, it is also possible that the test and validation error are *optimistic*. This often occurs when training and test data are coincident (i.e. in close proximity) to the training data (*Snee*, 1977), and results in an artificially low error. The danger of an optimistic assessment of model performance is that the model passes validation, but is highly likely to fail during deployment when it is challenged with previously unseen data.

In order to determine the relative performance of the different data splitting algorithms, a bootstrap evaluation of the data splitting algorithms was undertaken, following the methodology described in *Twomey and Smith* (1998). In a single test, the available data were split, using a given algorithm, into datasets with proportions of 3:1:1 (or, 60%, 20% and 20%) for training, test and validation, respectively. A generalised regression neural network (GRNN) (*Specht*, 1991) was trained to minimise the test error, and then the validation data were queried with the trained network. For each trained GRNN network, the test and validation performance were measured based on the mean squared-error (MSE), mean absolute error (MAE), and Pearson R^2 of predictions. This procedure was repeated for a total of 100 independent data splits. From the bootstrap results of each performance measure E, the average bias in test performance was determined as:

$$\bar{E}_{\mathcal{S}} = \frac{1}{100}\sum_{m=1}^{100} E_{\mathcal{S}}(m) \tag{9.2}$$

and the standard deviation in test performance $S(E_{\mathcal{S}})$ was determined as

$$S(E_{\mathcal{S}}) = \sqrt{\frac{1}{99}\sum_{m=1}^{100} (E_{\mathcal{S}}(m) - \bar{E}_{\mathcal{S}})^2} \tag{9.3}$$

where $E_{\mathcal{S}}(m)$ is the test performance for a single instance of training and testing data. Similarly, the mean and variance of the validation performance $E_{\mathcal{V}}$ was also determined.

The suite of data splitting algorithms evaluated included: random uniform sampling, systematic stratified sampling, two variants of SBSS, DUPLEX and the newly proposed SOMPLEX algorithm described in Section 9.3. These algorithms are summarised in Table 9.2. The first variant of SBSS was implemented with random allocation of a single datum per set from each map unit, with a map size of $k = n_{\mathcal{T}}$. The second variant of SBSS clustered the data on a SOM with $k = 5\sqrt{N}$ map units, where the Neyman allocation rule was used to determine the number of randomly selected data from each map unit. The Neyman allocation determines the number of points drawn according to (*Cochran*, 1977):

$$n_{\mathcal{T}}^{(m)} = \frac{\sigma_m N_m}{\sum_{j=1}^{k} \sigma_j N_j} \frac{n_{\mathcal{T}}}{N}, \tag{9.4}$$

where σ denotes the intra-cluster standard deviation. Neyman allocation improves random sampling of the SOM by increasing the sampling rate in highly populated clusters and where data are widely spread to ensure that more training samples are drawn from heterogeneous clusters (*May et al.*, 2009b).

Table 9.2: Algorithms included in the comparative study

Algorithm	Description
Random	Uniform random sampling.
Systematic	Data sorted on output axis prior to systematic sampling with interval $k_{\mathcal{T}} = n_{\mathcal{T}}/N$ for training data, and $k_{\mathcal{T}} = n_{\mathcal{S}}/N$ for test data.
SBSS(Single)	Random sampling of a single datum per map unit from a SOM with $k = n_{\mathcal{T}}$ map units.
SBSS(Neyman)	Random intra-cluster sampling of data from a SOM with $k = 5\sqrt{N}$, and where the Neyman allocation rule determines the number of training and test data drawn from each map unit.
DUPLEX	Duplex implementation of Kennard-Stone, as described by *Snee* (1977).
SOMPLEX	Multi-stage sampling with DUPLEX intra-cluster sampling on a SOM with $k = 2\sqrt{N}$ map units, as described in Section 9.3.

9.5 Datasets

Five datasets were sourced from water resources ANN modelling case studies presented within the literature: Colour, Turbidity, UV Absorbance (UVA), Salinity and Chlorine. These datasets, which are summarised in Table 9.3, represent various examples of ANN modelling applications. Each of the Colour, Turbidity and UVA datasets represent prediction tasks based on water treatment performance jar tests. The Salinity and Chlorine datasets represent two water quality forecasting applications using historical time-series data.

9.5.1 Pre-processing

Prior to undertaking the bootstrap evaluation, an input variable selection (IVS) stage was first applied as a pre-processing step in order to identify the important input variables in each dataset, and eliminate any redundant or irrelevant variables from the modelling and analysis. Each dataset was analysed using a forward selection algorithm based on partial mutual information (PMI), incorporating a termination criterion that minimises the Akaike Information Criterion (AIC) of the selected input variable set (see *May et al.* (2008a) for full details).

The resulting input and output variables for each modelling dataset are detailed in Table 9.3. The relative importance (RI) is also shown, and is measured as the PMI for each variable expressed as a percentage of the sum of PMI for all selected variables, which represents the total explanatory information within the input variable set (*Soofi and Retzer*, 2003). The relative importance also provides useful information in setting the parameters of the SOM, as mentioned in Section 9.3.

9.5.2 Coagulation

The Coagulation dataset was used previously in *Maier et al.* (2004) to demonstrate the potential for ANN prediction of treated water quality for a conventional surface water treatment process, comprising alum coagulation, flocculation, sedimentation and sand filtration, as determined by jar test simulation. The data comprise measurements of treated water quality corresponding to 204 individual jar tests performed over a range of alum doses for several different raw water sources.

Six potential input variables in the dataset included raw water alkalinity, pH, turbidity, UVA ($\lambda = 254$nm), colour and applied alum dose. Three modelling tasks are considered in this study, which are to predict colour, turbidity and UVA of the filtered water. Individual models were used to predict each filtered water quality variable, and since the input variables differ for each output, three modelling datasets were generated by repeating the input variable selection procedure for each output variable. These datasets are referred to as the Colour, Turbidity and UVA datasets, according to their respective output variables.

9.5.3 Salinity

The Salinity dataset has been used as a case study system to illustrate the application of ANN models to forecasting water quality in the River Murray, South Australia (*Maier and Dandy*, 1997; *Bowden et al.*, 2002). In this case study, the ANN modelling task is to generate a 14-day forecast of salinity at a downstream location using observations of water quality, and stream flow at upstream locations. The dataset comprises a total of 2028 weekly observations of stream flow and salinity at several upstream locations that are routinely sampled. The original data comprised a total of twelve variables, with up to 26-week lags of each variable, resulting in a total of 416 potential model inputs. However, the IVS procedure reduce this to an optimal set of only two input variables.

Table 9.3: Summary of modelling datasets used for the comparative study of data splitting algorithms

Dataset	Description	Variables	PMI	RI (%)
Colour	Multivariate; skewed, non-uniform and sparse distribution	Filtered water colour[a], HU	-	-
		Raw water UVA, cm^{-1} @ 254nm	0.278	28.3
		Raw water alkalinity, mg/L	0.217	22.1
		Alum dose, mg/L	0.181	18.5
		Raw water colour, HU	0.122	12.4
		Raw water pH	0.101	10.3
Turbidity	Multivariate; skewed, non-uniform and sparse distribution	Filtered water turbidity[a], NTU	-	-
		Raw water colour, HU	0.193	30.7
		Raw water turbidity, NTU	0.177	28.2
		Alum dose, mg/L	0.120	19.1
		Raw water UVA, cm^{-1} @ 254nm	0.079	12.6
		Raw water pH	0.059	9.4
UVA	Multivariate; skewed, non-uniform and sparse distribution; single dominant input	Filtered water UVA[a], cm^{-1} @ 254nm	-	-
		Raw water UVA, cm^{-1} @ 254nm	0.553	66.3
		Raw water alkalinity, mg/L	0.139	16.7
		Alum dose, mg/L	0.134	16.1
		Raw water pH	0.008	1.0
Salinity	Low-dimensional; smooth, dense distribution; single dominant input	14-day Murray Bridge salinity forecast[a], mg/L	-	-
		Mannum salinity, mg/L	1.268	81.5
		Waikerie salinity, mg/L	0.287	18.5
Chlorine	Moderately multivariate; smooth, dense distribution; single dominant input	24-hour trunk main free chlorine forecast[a], mg/L	-	-
		Trunk main free chlorine (t), mg/L	1.087	78.3
		Temperature (t-13), degC	0.106	7.6
		Filtered water storage free chlorine (t), mg/L	0.105	7.6
		Trunk main free chlorine (t-24), mg/L	0.090	6.5

[a] denotes output variables

9.5.4 Chlorine

The Chlorine dataset comprises 2773 observations of hourly water quality and flow within the Myponga water distribution system in South Australia, recorded for the period November 2003 and July 2004. This case study has been described previously in *May et al.* (2008b) and *Bowden et al.* (2006). The task in this application is to generate a 24-hour forecast of free chlorine within the trunk main, based on previous values at the downstream location, upstream water quality and system flows. Given up to 48-hour lags of the individual time-series for each variable, the dataset contained a total of 384 candidate input variables. In the study by *May et al.* (2008b), the IVS procedure was applied during model development to reduce the dataset to four input variables, and so the same set of input variables were retained for the purpose of this study.

9.6 Results and Discussion

The results of the comparative study are presented in Table 9.4, which summarises the biases and standard deviations of model test and validation performance using different data splitting algorithms when applied to each of the datasets. Based on the results obtained, the issue of variance in model performance when using random data splitting is evident. In the case of the Colour, Turbidity, and UVA datasets, variation in model performance was found to be significant when random data splitting was performed, leading to a high average error. In other words, the poor expected performance was driven by the high degree of variability in model performance. Models built using a random split of the Salinity dataset performed relatively well, in terms of average performance. However, the large standard deviation indicated the potential for this data splitting approach to yield poor test and validation performance, even for these datasets. For the Chlorine dataset, random sampling compared reasonably well with the other data splitting approaches. This dataset consisted of a large number of data sampled over a low-dimensional input space, with a single dominant input variable, and a distribution of data that was close to Gaussian. Not surprisingly, the impact of data splitting was therefore found to be negligible in this case.

The systematic stratified approach, in which data were first sorted along the output variable and then systematically sampled, was also found to yield poor performance for all but the Chlorine datasets. Although the approach is classified as deterministic, the variability introduced by the randomised location of the first

Table 9.4: Performance of data splitting algorithms on the water resources datasets

Algorithm	Set[a]	Colour			Turbidity			UVA			Salinity			Chlorine		
		RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2	RMSE	MAE	R^2
Random	$\mathcal{S}$	3.14 (2.25)[b]	2.10 (0.43)	0.847 (0.071)	0.434 (0.442)	0.197 (0.071)	0.716 (0.245)	0.080 (0.110)	0.020 (0.000)	0.816 (0.259)	16.5 (8.3)	10.4 (1.1)	0.997 (0.000)	0.066 (0.026)	0.040 (0.000)	0.981 (0.000)
	$\mathcal{V}$	3.50 (2.69)	2.22 (0.45)	0.806 (0.089)	0.544 (0.505)	0.224 (0.071)	0.613 (0.283)	0.085 (0.118)	0.020 (0.000)	0.760 (0.346)	21.5 (61.1)	11.6 (1.8)	0.995 (0.000)	0.070 (0.037)	0.041 (0.000)	0.978 (0.000)
Systematic	$\mathcal{S}$	2.98 (1.77)	2.01 (0.27)	0.857 (0.045)	0.509 (0.290)	0.243 (0.055)	0.472 (0.326)	0.074 (0.095)	0.019 (0.000)	0.820 (0.265)	16.8 (8.6)	10.4 (1.4)	0.997 (0.000)	0.066 (0.019)	0.038 (0.000)	0.981 (0.000)
	$\mathcal{V}$	3.17 (1.31)	2.13 (0.28)	0.830 (0.032)	0.626 (0.385)	0.245 (0.063)	0.881 (0.045)	0.020 (0.000)	0.013 (0.000)	0.950 (0.000)	18.7 (41.9)	12.3 (1.9)	0.997 (0.000)	0.065 (0.015)	0.038 (0.000)	0.981 (0.000)
DUPLEX	$\mathcal{S}$	3.66 -	2.39 -	0.804 -	0.232 -	0.095 -	0.978 -	0.018 -	0.013 -	0.918 -	20.2 -	13.3 -	0.996 -	0.076 -	0.046 -	0.966 -
	$\mathcal{V}$	1.91 -	1.49 -	0.750 -	0.341 -	0.000 -	0.938 -	0.021 -	0.018 -	0.904 -	21.1 -	13.8 -	0.996 -	0.072 -	0.045 -	0.970 -
SBSS (Single)	$\mathcal{S}$	3.37 (2.08)	2.35 (0.32)	0.810 (0.055)	0.447 (0.324)	0.193 (0.045)	0.860 (0.077)	0.075 (0.095)	0.020 (0.000)	0.791 (0.266)	20.2 (11.2)	13.8 (2.4)	0.995 (0.000)	0.082 (0.026)	0.053 (0.000)	0.963 (0.000)
	$\mathcal{V}$	4.09 (1.94)	2.74 (0.26)	0.807 (0.055)	0.472 (0.385)	0.198 (0.032)	0.684 (0.226)	0.105 (0.055)	0.028 (0.000)	0.542 (0.342)	65.2 (48.0)	23.0 (2.4)	0.964 (0.000)	0.106 (0.034)	0.062 (0.000)	0.955 (0.000)
SBSS (Neyman)	$\mathcal{S}$	2.81 (1.84)	1.88 (0.30)	0.768 (0.063)	0.219 (0.190)	0.122 (0.055)	0.720 (0.245)	0.067 (0.100)	0.018 (0.000)	0.849 (0.167)	17.8 (9.3)	10.7 (1.3)	0.997 (0.000)	0.065 (0.025)	0.039 (0.000)	0.981 (0.000)
	$\mathcal{V}$	3.08 (2.09)	1.95 (0.33)	0.767 (0.089)	0.221 (0.190)	0.112 (0.055)	0.559 (0.283)	0.075 (0.100)	0.020 (0.000)	0.735 (0.352)	21.4 (26.1)	11.7 (1.2)	0.996 (0.000)	0.070 (0.027)	0.041 (0.000)	0.979 (0.000)
SOMPLEX	$\mathcal{S}$	1.92 -	1.58 -	0.939 -	0.215 -	0.119 -	0.934 -	0.017 -	0.011 -	0.965 -	18.6 -	12.2 -	0.996 -	0.068 -	0.037 -	0.981 -
	$\mathcal{V}$	1.47 -	1.15 -	0.933 -	0.200 -	0.112 -	0.770 -	0.016 -	0.010 -	0.960 -	19.6 -	12.2 -	0.997 -	0.060 -	0.035 -	0.982 -

[a] $\mathcal{S}$ and $\mathcal{V}$ denote test and validation data sets, respectively.

[b] Values in parentheses are the standard deviation

sample drawn was determined to be significant in some cases, such as the Salinity and Colour datasets.

SBSS with allocation of a single, randomly selected datum for training and testing data, was found to perform the worst of all data splitting approaches. The approach led to poor test performance and extremely poor validation performance, which was attributed to a lack of representative data. The poor performance can be attributed to the fewer data that were included in training and test sets, since although a SOM was specified with a number of map units equal to the desired number of training data ($k = N_{\mathcal{T}}$), the SOM contained some empty units, which reduced the amount of information contained in the training set, in comparison to the other data splitting approaches, where the intended 3:1:1 split was achieved. The large variance in performance also indicates that sampling a single datum from within each cluster is insufficient and can result in a highly variable set of training data.

The drawback of random sampling within SOM partitions can be considered by visualising the distribution of data and the true shape of the partitions within the multivariate data space. Figure 9.2 describes the partitioning of the data by the 1×90 SOM for each plane formed by the given pairs of variables shown. The square markers indicate the location of the codebook vectors of the trained SOM. The delineated regions are the *Voronoi* tesselations that form the boundaries of each of the partitions formed by each codebook vector. These plots emphasize the characteristic behaviour of the SOM, which maps the density of the database $\mathcal{D}$. The resulting Voronoi regions vary in size according to the density of the data. In the dense regions, the partitions are small in volume and close together. In the sparse regions, the partitions are larger in volume, and the data within these partitions can be spread significantly.

Figure 9.2 also highlights deficiencies with some of the other potential data selection approaches when combined with the SOM clustering. For example, choosing training and test data closest to the codebook can ignore points within the cluster and can therefore be significantly biased, since the codebook vectors do not always correspond to the centroid of each of the partitions. In fact, choosing just a single point at random will not fully represent the data that are contained within a single partition. The need to ensure adequate coverage of the more voluminous partitions is immediately apparent.

SBSS using Neyman allocation, in which the number of training data drawn increases proportionally with spread and size of the cluster, was found to marginally improve on random uniform sampling. The average error was lower than for random uniform sampling in the Coagulation datasets. This improvement was de-

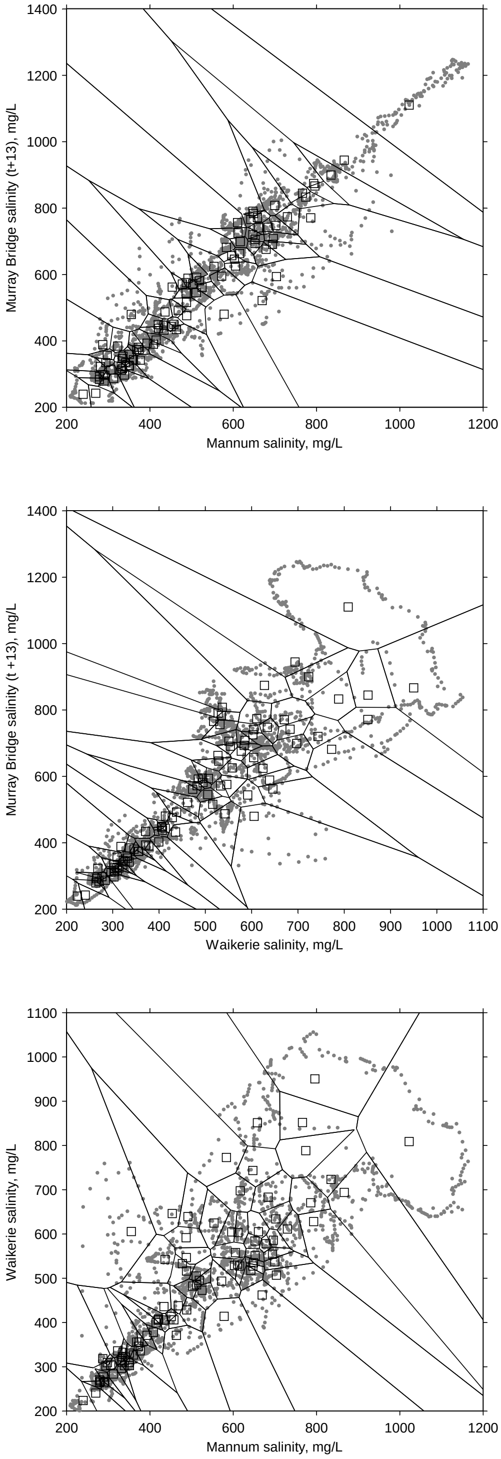


Figure 9.2: Codebook vectors and resulting Voronoi regions for the partitioning of the Salinity dataset by a 1×90 SOM

termined to be the effect of reduced variance in model performance, indicating that the approach could be used to sample representative data with increased reliability. This result was consistent with previous comparisons made by *May et al.* (2009b) using an example regression problem. However, in some instances, SBSS (Neyman) still performed poorly, indicating that the randomness of intra-cluster sampling still offered the potential for an occasional bad split.

The DUPLEX data splitting approach is fully deterministic, and generates a unique split for any given dataset, so the split was performed only once, hence there was no variability for these cases. Comparisons of the error obtained for each dataset showed that the DUPLEX algorithm was found to yield improved sampling for the Turbidity and UVA datasets, and a marginally pessimistic model error when applied to the Salinity and Chlorine datasets. In the case of the Colour dataset, the test and validation errors were significantly higher than the average error for random data splitting. This was found to be due to the forced inclusion of an outlying point in the validation data, which can occur as a result of the sampling behaviour. The underlying issue in this case is the sparseness of the data, and the result suggests a need for further data collection to provide more neighbouring data for training and testing.

Figure 9.3 provides a scatter plot indicating the selected training, testing and validating data when the SOMPLEX approach was used in conjunction with the partitioning shown in Figure 9.2. Training data are indicated by dot markers, test data are the filled square markers, and validating data are the unfilled square markers. It can be seen in these plots that the DUPLEX sampling within each of the SOM partitions is able to adequately sample representative data within each region, regardless of the volume and spread of data within a given partition.

In comparison with the other data splitting approaches evaluated, SOMPLEX was found to perform favourably, with low test and validation error in nearly all cases, with the exception of the Salinity dataset. Furthermore, when using this approach, the test and validation error were found to be consistent, indicating that good similarity between test and validating data sets was achieved. The clustering ensured that outlying data in each of the distinct regions identified by the SOM were first included in training, which can improve the quality of training, thereby improving upon the excessive pessimism of the conventional DUPLEX approach.

In this study, the number of map units of the SOM used to implement SOMPLEX was specified as $k = 2\sqrt{N}$, which was found to give good results. Previous experiments with SBSS (Neyman) found that the performance of SBSS improved for a larger grid $k = 5\sqrt{N}$ (*May et al.*, 2009b). This is because a larger map gen-

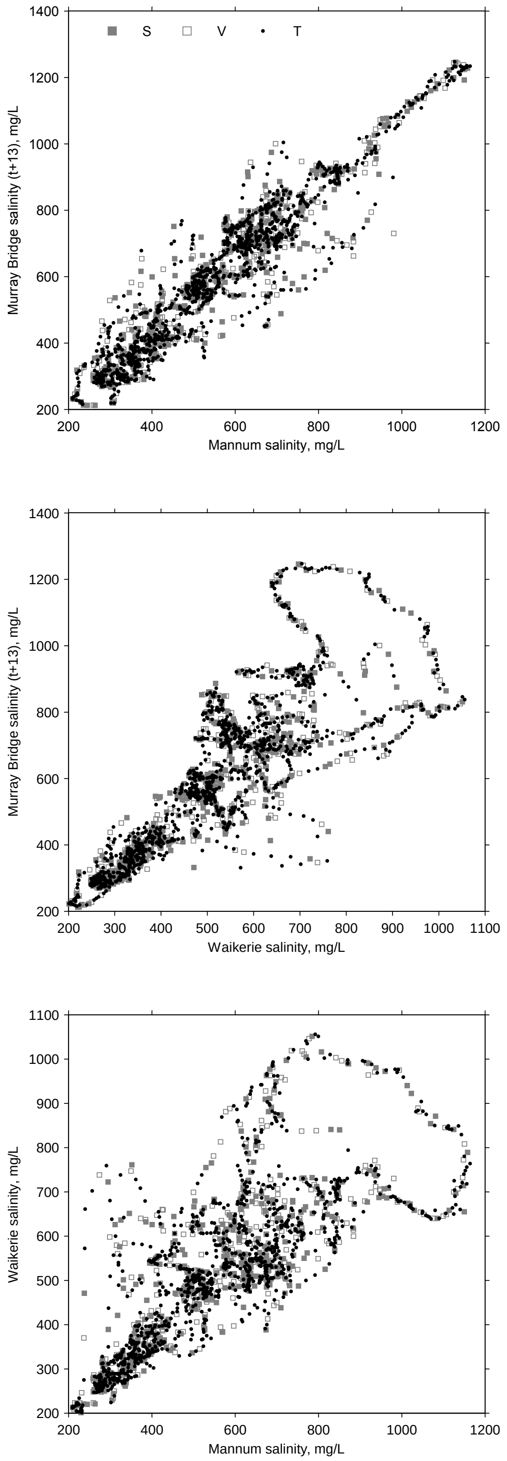


Figure 9.3: Training, test and validating data selected from the Salinity dataset using SOMPLEX

erates smaller, more homogeneous partitions, which reduces the variance of the random intra-cluster sampling. However, SOMPLEX can accommodate a coarser partitioning of the data, since the DUPLEX intra-cluster sampling provides good coverage of the partitions.

A common issue in all applications of the SOM is the treatment of singleton clusters (map units with a single datum). In data splitting applications of the SOM, the data in singleton clusters are typically allocated to training. The argument for doing so is that ANN models are poor at extrapolating, and consequently these data should be placed inside the training set to avoid the need to extrapolate (*Bowden et al.*, 2002). However, as the SOM size increases, the number of singleton clusters increases and the selection of data in these regions can be highly biased towards the training set. Fewer test and validation data are drawn, and are drawn from the more heavily populated clusters, which correspond to dense regions in the database. The impact of specifying a large grid is to increase the optimism with respect to test and validation performance. Although the formation of singletons is dependent on the presence of outliers within the data, the SOM map size should be restricted in order to avoid the unnecessary formation of singleton clusters caused by an overly large map. In this study, the $2\sqrt{N}$ was found to provide a suitably conservative map size.

9.7 Software

A suite of command-line data splitting tools has been developed, which allow users to implement all of the sampling approaches described in this paper. The software is freely available upon request for research and teaching purposes. The suite has been developed in C++ and has been compiled for operation on both Unix and Windows operating systems.

9.8 Conclusions

Data splitting is an important aspect of ANN development and ensures that good generalisation is achieved through cross-validation during training, and validation of model selection. Many methodologies have been considered for implementing data splitting, however in many cases the potential bias and variance in model performance due to the sampling procedure are neglected.

A comparative study was undertaken using several real-world ANN applications, including modelling of a coagulation process to predict turbidity, colour and UVA-254 of the treated water; time-series forecasting of salinity within a river; and forecasting of free chlorine residual within a water distribution system. This study has provided a useful evaluation of the bias and variance in model performance that results when the proposed SOMPLEX data splitting algorithm is used, relative to several other popular data splitting approaches. In the case of all but one the examples given in this paper, a lower model error was achieved using the SOMPLEX approach, with no variation. The benefit of the approach was greatest for the Coagulation datasets, which were characteristically multivariate with non-uniformly distributed data. For univariate or bivariate and smoothly distributed data, such as the Salinity and Chlorine datasets, the benefits over random sampling were found to be marginal. However, the results suggest that either SOMPLEX or DUPLEX can be universally applied with greater confidence than techniques such as systematic sampling, random uniform sampling or stratified random sampling.

The weakness of uniform random sampling was found to be a large variation in model performance for independent resampling of the data into training, test and validating sets. The large variance resulted in a high expected test and validation error when using this approach, which creates a high degree of uncertainty regarding the assessment of an ANN developed from a one-off split. This result suggests a lack of confidence in model performance resulting from data splitting using a random sampling approach for real-world datasets, which are often non-uniformly distributed and sparse.

A modified form of the DUPLEX algorithm has been introduced for sampling three datasets for training, testing and validating ANN models. DUPLEX provides a robust approach to data splitting, since the sample is guaranteed to cover the entire database volume, which avoids the need for extrapolation, and ensures representative data are selected in each set, regardless of the distribution of the data. However, the computational requirement of DUPLEX scales poorly with increasing dataset length, due to its operational complexity, which limits its potential application to splitting large datasets.

Conceptually, the SOM is useful at determining representative regions within the database. However, in practical terms, the approach is difficult to implement. The method is sensitive to the SOM parameters, and random sampling of the potentially non-uniform clusters that are generated by the SOM can still result in poor results due to large intra-cluster variance. Despite several applications of the SOM to data splitting, the most appropriate implementation of the SOM, and the best method for the selection of data are difficult to determine. The results in this study found that the SBSS approach using Neyman allocation lowered variance in model performance, relative to uniform random sampling, but did not eliminate it entirely.

The newly proposed SOMPLEX algorithm provides a multi-stage solution that combines SOM clustering with the DUPLEX algorithm. The SOM allows a coarse partitioning of the data into disjoint sets, which span the entire database. The DUPLEX algorithm then ensures that the intra-cluster samples selected provide adequate coverage of the cluster, which provides a more robust approach than random sampling. Computational run-time is reduced, and run-time performance is maintained by exploiting the scalability of the SOM to large datasets, while maintaining data splitting quality. Additionally, using the SOM approach can provide increased flexibility to the overall DUPLEX approach, since it provides scope for boosting (creating replicates) where data are sparse, or the ability to adjust the sample quota from each cluster depending on the density mapping determined by the SOM.

Chapter 10

Development of Artificial Neural Networks for Water Quality Modelling and Analysis

Publication 6

Publication Details

This work has been published as a book chapter as follows:

> May, R. J., H. R. Maier, and G. C. Dandy. Development of Artificial Neural Networks for Water Quality Modelling and Analysis, in G. Hanrahan (Ed.), *Modelling of Pollutants in Complex Environmental Systems*, Vol. 1, pp. 27–62, ILM Publications, UK, *2009*

Although the manuscript has been reformatted in accordance with University guidelines, and sections renumbered for inclusion within this thesis, the material within this paper is otherwise presented herein as published.

Statement of Authorship

May, R. J. (Candidate)
Development of ideas and criticisms, literature review, preparation of manuscript and corresponding author.

Signed: .... Date: ....

Maier, H. R.
Assistance with manuscript preparation and review of manuscript.

Signed: .... Date: ....

Dandy, G. C.
Review of manuscript.

Signed: .... Date: ....

May, R. J., Maier, H.R. and Dandy, G.C., (2009) Development of Artificial Neural Networks for Water Quality Modelling and Analysis, in *Modelling of Pollutants in Complex Environmental Systems, v. 1,* ed. G. Hanrahan, ILM, St. Albans, pp. 27–62.

NOTE: This publication is included on pages 259-299 in the print copy of the thesis held in the University of Adelaide Library.

Chapter 11

Conclusions

> *"If computers get too powerful, we can organize them into a committee—that will do them in."*
>
> Anonymous, *Bradley's Bromide*

Modelling is becoming an increasingly important tool, as the issues surrounding water resource management are becoming more complex and challenging. The development of models that can accurately and reliably represent the complexities of natural environmental systems and man-made infrastructure is essential for making informed decisions, and developing sound management practices. Artificial neural networks (ANNs) provide a powerful tool for the development of data-driven statistical models where processes are potentially non-linear and relatively poorly understood. Interest in the application of ANN models to environmental modelling and analysis has been growing exponentially in recent years, based on the numbers of reported applications. However, a majority of papers are concerned with the novelty of ANN models for a given application. Somewhat less attention is paid to the ANN model development approach that is employed. Many examples of *ad hoc* model development can be found within the literature, with many modelling decisions relying on case-specific idiosyncrasies or expert judgement. A more consistent ANN methodology is required, which will enable modellers to make informed decisions regarding ANN development. Consequently, this thesis has addressed the issue of how to develop ANN models in a structured and methodical fashion.

11.1 Contributions of Research

Overall, the major contribution of this research has been to provide further insight into the underlying development of ANN models, to guide model developers through the ANN development framework. Ultimately, it is hoped that this will encourage the adoption of a framework that leads to reliable model development and increased confidence in the ANN methodology. More specifically, the major efforts of this research have led to significant contributions in the input variable selection (IVS) and data splitting stages of ANN development.

11.1.1 Input Variable Selection

The contributions have been made in the area of input variable selection, which relate directly to the research objectives defined in Section 1.3, are summarised as follows:

1. The IVS problem is common to all statistical modelling applications, where a number of potential predictors are available, but it is necessary to include only the most informative variables, and discard irrelevant and redundant variables. Determining what IVS approach is most suitable for ANN development is a challenge for practitioners, given the many different algorithms that are proposed within the literature. A taxonomical representation produced from the review of literature provides a useful reference for classifying the many approaches that have been described. Comparisons of the strengths and weaknesses the different IVS algorithms are made with respect to their application to ANN model development. Model specific wrapper algorithms can be applied, but are likely to yield case-specific results, that are a function of the particular ANN class, learning algorithm and data. Model-free filter methods, which are based on statistical relevance measures, tend to provide a faster and more generic approach. Many IVS methods are employed that are based upon linear modelling paradigms, such as correlation and covariance analysis. However, the assumption with ANN model development is that some relationships are non-linear, and consequently linear analysis can fail to identify important variables. Mutual information (MI) is preferable for ANN development, as it measures arbitrary relationships, and provides a suitable alternative relevance measure for implementing IVS.

2. An evaluation of a forward selection procedure that estimates partial mutual mutual information (PMI) was undertaken using a suite of benchmark

datasets. The PMI-based algorithm requires a bootstrap estimate of the critical value of MI for a given sample size, in order to determine the significance of PMI during each iteration. This significance test forms the basis of the termination criterion, which halts the selection procedure. The bootstrap is necessary, because no method currently exists for directly computing the confidence bounds of MI estimates. However, estimating MI is computationally intensive and, due to computational constraints, a limited number of bootstrap replicates are used to determine the critical value of MI. This results in inaccurate and highly variable estimates and the termination criterion was found to be inaccurate and unreliable. A comparison with a similarly devised forward-selection algorithm using linear correlation was also undertaken in parallel. The results of this comparison clearly demonstrated the inability of correlation-based IVS to fully identify the input variable set when one or more input-output relationships are non-linear. This research has therefore conclusively shown the benefit of MI, which had not previously been fully tested.

3. An improved algorithm was developed that provides a fast and accurate approach to performing input variable selection (IVS) based on the estimation of partial mutual information (PMI). Specifically, several alternative termination criteria were developed that sidestep the requirement to perform the computationally intensive bootstrap at each iteration of the PMI-based forward selection procedure. Off-line estimates of critical values were estimated using Monte Carlo simulation. These critical values provide an immediate threshold to gauge the significance of PMI for a given candidate input variable. However, the derived estimates of critical values assume i.i.d Gaussian data, which may not be a reasonable assumption for real-world datasets. Alternatively, the Akaike Information Criterion (AIC) can be computed, to determine an optimum trade-off between the number of selected variables, and the information contained within the input set. A third alternative criterion was determined that applies an outlier test to determine whether the remaining candidates contain any significantly relevant variables. This approach was found to suffer from the effect of masking, which is a known issue in outlier detection, where the relevance of variables is masked by distribution multiple relevant variables. However, methods to overcome this problem were identified, including modifications to the outlier test that is used, or by supplementing the candidate input variables with sufficient noise variables. Overall, the result of this research has improved the accuracy of selections, and has significantly reduced the computational requirement that is necessary to perform the task using the PMI-based approach.

4. The benefits of the IVS approach using the PMI-based forward-selection algorithm were demonstrated using a meta-modelling example and a real-world

example of disinfectant forecasting. In each case, the IVS approach was able to reduce a large number of potential input variables into a smaller, more parsimonious subset, which contained all of the information necessary to represent the behaviour of the output variable. From a statistical perspective, the quality of the smaller model was considered superior to larger models with a greater number of input variables, given the tendency for large ANN models to be over-fit. In a practical sense, the smaller model utilised less data and could be developed faster due to the reduced computational requirements of a smaller dataset. The additional benefit of the IVS procedure is that it yields useful information regarding the importance of variables, as this can be directly measured based on the PMI estimated for each input variable. The importance can be used to infer the relative strength of input-output relationships that are embodied within ANN models, which may provide some useful explanation and validation of how model predictions are yielded. It may also provide a useful tool in troubleshooting unexpected predictions, which would not be easily facilitated by a "black-box" approach.

11.1.2 Data Splitting

The contributions of this research, with respect to methods used during the data splitting stage of ANN development, are summarised as follows:

1. The review of literature has highlighted the importance of data splitting for ANN development. The issues of generalisation are a key concern for ANN development, since the arbitrary complexity that ANN architectures afford also increases the potential to over-fit. Although a number of approaches can be employed to prevent over-fitting, the train-and-test method is by the far the most widely used, and it is ultimately necessary to set aside data for validation, regardless of the method used to determine the ANN parameters. A taxonomy of data splitting algorithms is presented, which organises the different approaches presented within literature. The key considerations relevant to ANN model development are the ability to reliably select training, test and validating data sets that lead to representative training data and an unbiased estimate of model performance. The most commonly employed methods within ANN literature are found to be random, and therefore subject to high variance; or, convenience sampling, which has a risk of generating highly biased datasets. Recently proposed algorithms based on the self-organizing map (SOM) are categorised into the class of stratified sampling algorithms. These are found to improve on random sampling, by reducing the variance of the data sample. However, there are several different ways in which the SOM

is used to perform the data sampling, and no evidence to suggest the most appropriate approach. The literature review also identified Kennard-Stone sampling algorithms (CADEX and DUPLEX), which have not been used widely in environmental modelling and analysis applications. These algorithms are deterministic, and therefore result in zero sample variance, and select representative data for each dataset regardless of the distribution of the data.

2. The application of the SOM to data splitting was examined in detail. Experiments were conducted on an example regression problem to assess the quality of data splitting using different allocation rules, and different SOM sizes. The results indicated that the variation in sample performance was reduced when the Neyman allocation rule was adopted, which increases the sampling rate in proportion to the size and spread of a SOM cluster. The common approach of using a large SOM and drawing one datum each for training and testing was found to yield poor results.

3. How to correctly size the SOM has been an on-going issue for all applications of the SOM, and no reliable guidelines for determining an appropriate size had been previously developed. In this research, the issue of SOM size was examined directly by considering the relationship between SOM size and the performance of the SOM-based data splitting algorithm. The use of cluster validity measures was found to be inappropriate for determining the most appropriate size of the SOM for data splitting. Cluster validity indices (CVIs) are useful for identifying the number of natural clusters, but data in ANN modelling are not always strongly clustered, and these measures therefore tend to suggest a small SOM. However, an analysis of grid size found that the best sampling was obtained for a moderate to large size grid. This is was found to be consistent with the formation of smaller, more homogeneous partitions by the SOM, rather than the identification of large clusters. A heuristic rule for specifying the SOM size was found to result in a size that approximately corresponded to optimal sampling, and that this could be used instead to determine the size SOM required to partition a given dataset, without the need for extensive trial-and-error.

4. A novel algorithm was proposed that combines the SOM with DUPLEX intra-cluster sampling. The weakness of the SBSS approach is that the random intra-cluster sampling still has the potential to result in a highly variable split, since the data in some SOM partitions can be significantly large and spread. Using DUPLEX, the data within each SOM partition can be more representatively split into training, test and validating sets. This improves the sampling of partitions, and eliminates the variance that results from random sampling. Furthermore, DUPLEX is a computationally intensive data splitting algorithm,

and scales poorly to large datasets. However, the SOMPLEX approach provides a significant reduction in the computational effort from $O\{N^3\}$ to $O\{N^{1.5}\}$, which allows the DUPLEX approach to be applied to much larger datasets.

5. The performance of data splitting approaches was compared for some real-world examples of the application of ANNs to water resources modelling. The results highlighted the potentially high degree of variability in model performance when random data splitting is used. The application of SOMPLEX and DUPLEX was demonstrated in each case to reliably perform an unbiased one-shot data split, with training, testing and validation datasets that were equally representative of the modelling database.

11.1.3 Water Quality Forecasting

A motivation for this research, from the perspective of water authorities, is the development of water quality forecasting capabilities, or predictive models that can describe water quality changes within a large-scale WDS. The ultimate goal is to determine a suitable model that can inform operators and assist in developing optimal water quality management practice. The outcomes of this research, with respect to the modelling of water quality within a WDS, are summarised as follows:

1. The real-world Myponga WDS case study that has been presented within Chapters 7 and 9 provides a successful demonstration of the ANN modelling approach to forecasting residual chlorine. The application of the ANN approach provides a validation of the model development framework, in particular the novel approaches for IVS and data splitting, within an applied context. In conjunction with the closely related case study examples presented in Chapter 9, this research has positively demonstrated the utility of ANN modelling within a range of applications that have direct relevance to water authorities.

2. The analysis of WDS dynamics, and the assessment of relevant input variables, was found to provide a useful insight that was gained through the application of the IVS approach. An interesting result was obtained that challenges results in previous studies, with respect to the influence of temperature as an input to ANN models. In the case of the Myponga WDS, the diurnal variation in demand gave rise to strong 24-hour periodicity in the chlorine trends. Similarly, water temperature also exhibited a strong 24-hour sinusoidal cycle, and was subsequently identified as a highly relevant input due to a strong correlation with the chlorine trend. Although highly correlated, it is doubtful that the relationship identified by the ANN can be considered a causal effect. Previous

studies have implied or suggested that chlorine decay kinetics are in some way influencing the observed chlorine decay. However, in this case, the inclusion of water temperature appears more to provide an input variable that allows the ANN to track the 24-hour cycle. In the absence of water temperature data, it is possible that the inclusion of surrogate variables could also be used to track seasonal patterns. These variables are synthetically generated data, where the data generating functions are cyclic functions with an appropriate period, such as sine and cosine functions. This approach has been used previously in ANN prediction of electricity demand, and could also be potentially useful for ANN prediction of water quality within a WDS.

3. Although accurate forecasting of residual chlorine has been demonstrated to be achievable using the ANN modelling approach, some insight gained through this work suggests that input-output modelling of large-scale water distribution networks is not likely to be as straightforward a task as previously thought. This is because the impacts of relatively small adjustments in chlorine dose at the outlet of the water treatment plant (WTP) are significantly dampened within the network, due to long detention times within trunk mains and within storage reservoirs and the natural decay of chlorine over time. In the case of the Myponga WDS, the variation in chlorine due to diurnal flow variation was found to be far more significant than variation caused by changes in applied dose. Consequently, the statistically relevant variables were identified as endogenous lags of chlorine at the control point within the WDS, rather than previous values of the chlorine residual at the outlet of the WTP, where the chlorine is dosed. This yields an accurate time-series forecasting ANN, but does not produce an input-output model that could be deployed to predict the impact of dose adjustments. However, it should also be noted that this limitation is in part due to the restricted variation in dose within the data collected, given that the system was under operational control during the data collection period, and that large variations in chlorine dose are avoided. As discussed in Section 11.2, this limits the ability of the ANN to learn the impact of large changes in chlorine dosing for this system, and restricts modelling to the prediction of deviations about a set-point due to disturbances in flows within the WDS.

4. The hypothetical case of the Cherry Hills—Brushy Plains WDS, presented in Chapter 7, provides an example of what could be possible using ANN models when applied to more complex networks. This case study describes the development of an ANN to model the relationship between the dose applied at multiple booster chlorination stations and the residual chlorine at a control point within a small, but complex network, that included a common inlet-outlet storage tank. In particular, this example showed that the IVS method

was able to identify the dynamics of the system over a 48-hour window, and identify the booster stations and the dosing window that specifically impacted on residual chlorine concentration at the control point.

11.1.4 Field Research

It is worth highlighting that the Myponga WDS case study, which is presented in Papers 3 and 5, represents a significant body of field-work that was undertaken to support this research. Initial establishment of robust field monitoring sites using multiple chlorine analysers, operation and calibration, communications and data retrieval were all tasks that were undertaken over a three-year period as part of this research in order to generate a dataset spanning three-years. The collected chlorine residual and temperature data were also collated with additional data obtained from the water utility to create a comprehensive modelling dataset for use in this research, and for use in future research. Given that robust and reliable data acquisition systems is essential to provide the amount of data required for ANN and other data-driven modelling applications, the practical know-how and experience gained from this aspect of the research project has immense value.

11.1.5 Software

A significant amount of this research has also involved the programming of software tools in order to develop and implement the various algorithms discussed within this thesis. In addition to the theoretical understanding, an additional outcome of this research is the implementation of ANN development software tools. A program has been developed to enable the selection of input variables using the PMIS algorithm described in Chapters 6 and 7. Several programs have also been developed to implement the various data splitting algorithms discussed in Chapters 8 and 10.

11.2 Research Limitations

In undertaking this research, time restrictions and other constraints have led to some limitations. The key limitations are summarised as follows:

1. The estimation of PMI is dependent on the ability to know or estimate the density distribution for the sample of available data. The work presented in this

thesis has focussed on the use of kernel density estimation (KDE) for the estimation of density distributions. This approach was chosen as it was relatively straightforward to implement, and was consistent with research in this area. However, KDE has some known limitations, including inaccuracy when applied to high-dimensional cases, and the need to determine the optimum kernel bandwidth. Alternative density estimation techniques may provide faster or more accurate estimates of PMI, and these would each provide a further improvement to the IVS algorithm.

2. Considerable attention in this research was given to the application of the SOM for performing the clustering of data, which has led to the SOMPLEX approach. However, the SOM is but one of many potential partitioning and hierarchical clustering techniques. Numerous multi-stage sampling approaches could be conceived by coupling DUPLEX intra-clustering sampling with any clustering algorithm, which may perform equally as well as SOMPLEX. Furthermore, improvements to the conventional SOM may also be explored, since any improvement in the clustering will ultimately improve the quality of the overall data splitting approach.

3. The generalised regression neural network (GRNN) was used as the ANN class in all of the studies undertaken, as the focus of the research was on the auxiliary stages of development, which were considered to be independent of the class of ANN used to perform the modelling. The GRNN was selected as it can be developed much faster than the more conventional MLP, and therefore allowed many bootstrap experiments to be undertaken within a reasonable time-frame. In particular, the performance of the MLP for bootstrap data splits may potentially yield different bias and variance characteristics than those of the GRNN.

Several practical limitations were also encountered in relation to ANN model development, which are given as follows:

1. Observability of a system is an important consideration during model development. In many cases, data for model development are sourced from readily collected datasets, such as SCADA or monitoring programs. The concern highlighted through the examples given in this research is that while there is an abundance of data, there is insufficient *information* within the dataset to describe the system. In particular, systems that are observed under closed-loop (controlled) conditions may not be excited to exhibit the full range of variation in the system response. It can not be expected that a model will be able to predict features that are not present in the data.

2. The particular example of disinfectant modelling highlighted some limitations in the potential for ANN-based control of water quality in large-scale distribution networks. The analysis of ANN models developed indicated that fluctuations in disinfectant were largely driven by diurnal patterns, which are caused by demand. Importance analysis of ANN inputs also reflected the weak relationships between downstream residual chlorine and post-injection chlorine concentrations. Such a result is symptomatic of control of a large scale process, where downstream responses to fluctuations in chlorine injection rates are dampened by mixing and decay over long detention times. The scope to perform process identification on water supply networks, by performing moderate adjustments of the chlorine injection rate, is likely to be restricted, as water authorities will be cautious of exceeding limits for residual at locations within close proximity to the treatment plant.

11.3 Future Research

Several opportunties for continuing research are identified:

1. The development of ANN models still remains a relatively unexplored alternative in many applications, and further work is required to investigate the potential contribution of ANNs. Additionally, the model development framework proposed in this thesis could be retrospectively applied to improve the quality of ANN models used in applications where they have already demonstrated some benefit. Such work would also continue to popularise the ANN approach and increase its acceptance within the mainstream water resources modelling community.

2. Data collection and measurement is important for any modelling application. Current trends in data acquisition are increasingly based on real-time sensing and communications, which offers the potential to gather vast amounts of data at a relatively low cost. However, without the tools necessary to turn *data* into *information*, there is little benefit and motivation for the development of these sensors. It is in this context that ANNs can potentially add significant value to such data acquisition systems. By the same token, ANN models are reliant on robust, accurate and affordable measurement to generate informative datasets that allow them to represent system behaviour. It is therefore logical that the development of reliable and affordable monitoring and data acquisition systems should go hand-in-hand with the development of computational tools for data analysis.

3. The software developed during this research has been developed purely for research purposes. Future work should also consider the further development of the techniques developed during this research within a user-friendly software application. Lack of software is a major barrier to the uptake of many of the newer developments in ANN development, and would provide a suitable alternative to existing tools that implement outmoded model development techniques.

References

Adams, D., Predicting the future, in *Salmon of Doubt: Hitchhiking the Galaxy One Last Time*, p. 282, Macmillan/Pan, UK, 2002.

Adeloye, A., The relative utility of regression and artificial neural networks models for rapidly predicting the capacity of water supply reservoirs, *Environmental Modelling & Software*, *24*, 1233–1240, 2009.

Akaike, H., A new look at the statistical model identification, *IEEE Transactions of Automatic Control*, *19*, 716–723, 1974.

Al-Alawi, S., S. Abdul-Wahab, and C. Bakheit, Combining principal component regression and artificial neural networks for more accurate predictions of ground-level..., *Environmental Modelling and Software*, *23*, 396–403, 2008.

Alp, M., and H. Cigizoglu, Suspended sediment load simulation by two artificial neural network methods using hydrometeorological data, *Environmental Modelling and Software*, *22*, 2–13, 2007.

Anctil, F., and N. Lauzon, Generalisation for neural networks through data sampling and training procedures, with applications to stream flow predictions, *Hydology and Earth System Sciences*, *8*, 940–958, 2004.

Andrews, R., J. Dieterich, and A. B. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, *Knowledge-Based Systems*, *8*, 373–389, 1995.

Ani, E.-C., S. Wallis, A. Kraslawski, and P. S. Agachi, Development, calibration and evaluation of two mathematical models for pollutant transport in a small river, *Environmental Modelling & Software*, *24*, 1139–1152, 2009.

Back, A. D., and T. P. Trappenberg, Selecting inputs for modeling using normalized higher order statistics and independent component analysis, *IEEE Transactions on Neural Networks*, *12*, 612–617, 2001.

Battiti, R., Using mutual information for selecting features in supervised neural net learning, *IEEE Transactions on Neural Networks*, *5*, 537–550, 1994.

Baxter, C. W., S. J. Stanley, and Q. Zhang, Development of a full-scale artificial neural network model for the removal of natural organic matter by enhanced coagulation, *Journal of Water Science Research Technology - Aqua*, *48*, 129–136, 1999.

Baxter, C. W., S. J. Stanley, Q. Zhang, and D. W. Smith, Developing artificial neural network process models: a guide for drinking water utilities, in *6th Environmental Engineering Society Specialty Conference of the CSCE*, vol. 376-383, Ontario, 2000.

Baxter, C. W., Q. Zhang, S. J. Stanley, R. Shariff, R.-R. T. Tupas, and H. L. Stark, Drinking water quality and treatment: the use of artificial neural networks, *Canadian Journal of Civil Engineering*, *28*, 26–35, 2001.

Bellman, R., *Adaptive control processes: a guided tour*, Princeton University Press, New Jersey, 1961.

Blum, A., and P. Langley, Selection of relevant features and examples in machine learning, *Artificial Intelligence*, *97*, 245–271, 1997.

Boccelli, D. L., M. E. Tryby, J. G. Uber, L. A. Rossman, M. L. Zierolf, and M. M. Polycarpou, Optimal scheduling of booster disinfection in water distribution systems, *Journal of Water Resources Planning and Management, ASCE*, *124*, 99–111, 1998.

Bonnlander, B. V., and A. S. Weigend, Selecting input variables using mutual information and nonparametric density estimation, in *International Symposium on Artificial Neural Networks*, pp. 42–50, Taiwan, 1994.

Bowden, G. J., Forecasting water resources variables using artificial neural techniques, Ph.d, University of Adelaide, 2003.

Bowden, G. J., H. R. Maier, and G. C. Dandy, Optimal division of data for neural network models in water resources applications, *Water Resources Research*, *38*, 1–11, 2002.

Bowden, G. J., G. C. Dandy, and H. R. Maier, Input determination for neural network models in water resources applications. part 1 - background and methodology, *Journal of Hydrology*, *301*, 75–92, 2005.

Bowden, G. J., J. B. Nixon, G. C. Dandy, H. R. Maier, and M. Holmes, Forecasting chlorine residuals in a water distribution system using a general regression neural network, *Mathematical and Computer Modelling*, *44*, 469–484, 2006.

Box, G. E. P., and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*, Holden-Day Inc., San Francisco, 1976.

Brasquet, C., and P. Le Cloirec, Qsar for organics adsorption onto activated carbon in water : What about the use of neural networks?, *Water Research*, *33*, 3603–3608, 1999.

Brent, R. P., *Algorithms for minimization without derivatives*, Prentice-Hall, 1973.

Broad, D., G. C. Dandy, and H. R. Maier, Water distribution system optimization using metamodels, *Journal of Water Resources Planning and Management, ASCE*, *131*, 172–180, 2005.

Carreira-Perpinan, M. A., A review of dimension reduction techniques, *Tech. rep.*, Dept. of Computer Science, University of Sheffield, 1997.

Caudill, M., Grnn and bear it, *AI Expert*, *8*, 28–33, 1993.

Cereghino, R., and Y.-S. Park, Review of the self-organizing map (som) approach in water resources: Commentary, *Environmental Modelling & Software*, *24*, 945–947, 2009.

Charef, A., A. Ghauch, P. Baussand, and M. Martin-Bouyer, Water quality monitoring using a smart sensing system, *Measurement*, *28*, 219–224, 2000.

Chernick, M. R., *Bootstrap methods: a practitioner's guide*, John Wiley & Sons, New York, 1999.

Chow, T. W. S., and D. Huang, Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information, *IEEE Transactions on Neural Networks*, *16*, 213–224, 2005.

Cigizoglu, H. K., Estimation and forecasting daily suspended sediment data by multi-layer perceptrons., *Advances in Water Resources*, *27*, 185–195, 2004.

Cigizoglu, H. K., Application of the generalized regression neural networks to intermittent flow forecasting and estimation., *ASCE Journal of Hydrologic Engineering*, *10*, 336–341, 2005.

Cigizoglu, H. K., and O. Kisi, Methods to improve the neural network performance in suspended sediment estimation, *Journal of Hydrology*, *317*, 221–238, 2006.

Clark, R. M., and R. C. Haught, Characterizing pipe wall demand: Implications for water quality modeling, *Journal of Water Resources Planning and Management, ASCE*, *131*, 208–217, 2005.

Cochran, W. G., *Sampling Techniques*, Wiley, New York, 1977.

Cote, M., B. P. A. Grandjean, P. Lessard, and J. Thibault, Dynamic modelling of the activated sludge process: Improving prediction using neural networks, *Water Research*, *29*, 995–1004, 1995.

Coulibaly, P., F. Anctil, and B. Bobee, Daily reservoir inflow forecasting using artificial neural networks with stopped training approach, *Journal of Hydrology*, *230*, 244–257, 2000.

Cover, T. M., and J. A. Thomas, *Elements of information theory*, Wiley series in telecommunications, John Wiley & Sons, Inc., New York, 1991.

Craven, M. W., and J. W. Shavlik, Using neural networks for data mining, *Future Generation Computer Systems*, *13*, 211–229, 1998.

Damas, M., M. Salmeron, and J. Ortega, Anns and gas for predictive control of water supply networks, in *International Joint Conference on Neural Networks*, vol. 4, pp. 365–370, 2000.

Darbari, A., Rule extraction from trained ANN: A survey, *Tech. rep.*, Institute of Artificial Intelligence, Dept. of Computer Science, TU Dresden, 2000.

Darbellay, G. A., An estimator of the mutual information based on a criterion for independence, *Computational Statistics & Data Analysis*, *32*, 1–17, 1999.

Darsono, S., and J. Labadie, Neural-optimal control algorithm for real-time regulation of in-line storage in combined sewer systems, *Environmental Modelling and Software*, *22*, 1349–1361, 2007.

Daszykowski, M., B. Walczak, and D. L. Massart, Representative subset selection, *Analytica Chimica Acta*, *468*, 91–103, 2002.

David, F. N., Tables of the correlation coefficient, in *Biometrika Tables for Statisticians*, edited by E. S. Pearson and H. O. Hartley, vol. 1, 3rd ed., Cambridge University Press, Cambridge, 1966.

Davies, L., and U. Gather, The identification of multiple outliers, *Journal of the American Statistical Association*, *88*, 782–792, 1993.

Dawson, C. W., and R. L. Wilby, Hydrological modelling using artificial neural networks, *Progress in Physical Geography*, *25*, 80–108, 2001.

Dawson, C. W., R. J. Abrahart, A. Y. Shamseldin, and R. L. Wilby, Flood estimation at ungauged sites using artificial neural networks, *Journal of Hydrology*, *319*, 391–409, 2006.

de Bodt, E., M. Cottrell, and M. Verleysen, Statistical tools to assess the reliability of self-organizing maps, *Neural Networks*, *15*, 967–978, 2002.

Despagne, F., and D. L. Massart, Neural networks in multivariate calibration, *Analyst*, *123*, 157–178, 1998.

Ding, C., and H. Peng, Minimum redundancy feature selection from microarray gene expression data, *Journal of Bioinformatics and Computational Biology*, *3*, 185–205, 2005.

Dionisio, A., R. Menezes, and D. A. Mendes, Mutual information: a measure of dependency for nonlinear time series, *Physica A*, *344*, 2004.

Dorigo, M., and T. Sutzle, *Ant Colony Optimization*, MIT Press, Cambridge, USA, 2004.

Dutot, A., J. Rynkiewicz, F. Steiner, and J. Rude, A 24-h forecast of ozone peaks and exceedance levels using neural classifiers and weather predictions, *Environmental Modelling and Software*, *22*, 1261–1269, 2007.

El-Din, A. G., D. W. Smith, and M. G. El-Din, Application of artificial neural networks in wastewater treatment, *Journal of Environmental Engineering Science*, *3*, 81–95, 2004.

Fernando, T. M. K. G., H. R. Maier, and G. C. Dandy, Selection of input variables for data driven models: An average shifted histogram partial mutual information estimator approach., *Journal of Hydrology*, *367*, 165–176, 2009.

Finardi, S., R. De Maria, A. D'Allura, C. Cascone, G. Calori, and F. Lollobrigida, A deterministic air quality forecasting system for torino urban area, italy, *Environmental Modelling and Software*, *23*, 344–355, 2008.

Flood, I., and N. Kartam, Neural networks in civil engineering, I, Principles and understanding, *Journal of Computing in Civil Engineering*, *8*, 131–148, 1994.

Fodor, I. K., A survey of dimension reduction techniques, *Tech. rep.*, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 2002.

Fogelman, S., M. Blumstein, and H. Zhao, Estimation of chemical oxygen demand by ultraviolet spectroscopic profiling and artificial neural networks, *Neural Computing & Applications*, *15*, 197–203, 2005.

Friedman, J., Multivariate adaptive regression splines. technical report no. 102., *Technical report*, Laboratory for Computational Statistics, Department of Statistics, Stanford University, 1988.

Gathercole, C., and P. Ross, Dynamic training subsect selection for supervised learning in genetic programming, in *Parallel Problem Solving from Nature III*, edited by Y. Davidor, H.-P. Schwefel, and R. Manner, vol. 866, pp. 312–321, Springer-Verlag, Berlin, 1994.

Geman, S., E. Bienenstock, and R. Doursat, Neural networks and the bias/variance dilemma, *Neural Computation*, *4*, 1–58, 1992.

Gibbs, M. S., N. Morgan, H. R. Maier, G. C. Dandy, J. B. Nixon, and M. Holmes, Investigation into the relationship between chlorine decay and water distribution parameters using data driven methods, *Mathematical and Computer Modelling*, *44*, 485–498, 2006.

Gill, A. A., G. D. Smith, and A. J. Bagnall, Improving decision tree performance through induction- and cluster-based stratified sampling, in *Intelligent Data Engineering and Automated Learning*, pp. 339–344, Springer-Verlag, Exeter, UK, 2004.

Goebel, B., Z. Dawy, J. Hagenauer, and J. C. Mueller, An approximation to the distribution of finite sample size mutual information estimates, in *IEEE International Conference on Communications (ICC-05)*, Seoul, South Korea, 2005.

Goldberg, D. E., *Genetic algorithms in seach, optimization and machine learning*, Addison-Wesley, Boston, 1989.

Granger, C. W., M. E., and J. Racine, A dependence metric for possibly nonlinear processes, *Journal of Time Series Analysis*, *24*, 649–669, 2004.

Gunter, S., and H. Bunke, Validation indices for graph clustering, *Pattern Recognition Letters*, *24*, 1107–1113, 2003.

Guyon, I., and A. Elisseeff, An introduction to variable and feature selection, *The Journal of Machine Learning Reearch*, *3*, 1157–1182, 2003.

Halkidi, M., Y. Batistakis, and M. Vazirgiannis, On clustering validation techniques, *Journal of Intelligent Information Systems*, *17*, 107–145, 2001.

Harrold, T. I., A. Sharma, and S. Sheather, Selection of a kernel bandwidth for measuring dependence of hydrologic time series using the mutual information criterion, *Stochastic Environmental Research and Risk Assessment*, *15*, 310–324, 2001.

Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference and Prediction*, Springer Series in Statistics, Springer, New York, 2001.

Hsu, K. L., H. V. Gupta, X. G. Gao, S. Sorooshian, and B. Imam, Self-organizing linear output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis, *Water Resources Research*, *38*, 2002.

Hu, T. S., K. C. Lam, and S. T. Ng, River flow time series prediction with a range-dependent neural network, *Hydrological Sciences Journal–Journal Des Sciences Hydrologiques*, *46*, 729–745, 2001.

Huang, D., and T. W. S. Chow, Effective feature selection scheme using mutual information, *Neurocomputing*, *63*, 325–343, 2005.

Hutter, M., and M. Zaffalon, Distribution of mutual information from complete and incomplete data, *Computational Statistics & Data Analysis*, *48*, 633–657, 2005.

Iglesias, A., C. Dafonte, B. Arcay, and J. Cotos, Integration of remote sensing techniques and connectionist models for decision support in fishing catches, *Environmental Modelling and Software*, *22*, 862–870, 2007.

Imrie, C. E., S. Durucan, and A. Korre, River flow prediction using artificial neural networks: generalisation beyond the calibration range, *Journal of Hydrology*, *233*, 138–153, 2000.

Ionescu, A., and Y. Candau, Air pollutant emissions prediction by process modelling - application in the iron and steel industry in the case of a..., *Environmental Modelling and Software*, *22*, 1362–1371, 2007.

Izrailev, S., and D. K. Agrafiotis, Variable selection for QSAR by artificial ant colony systems, *SAR and QSAR in Environmental Research*, *13*, 417–423, 2002.

Jain, A., and S. Srinivasulu, Integrated approach to model decomposed flow hydrograph using artificial neural network and conceptual techniques, *Journal of Hydrology*, *317*, 291–306, 2006.

Jain, A., K. Sudheer, and S. Srinivasulu, Identification of physical processes inherent in artificial neural network rainfall–runoff models, *Hydrological Processes*, *18*, 571–581, 2004.

Kaboudan, M. A., Diagnosing chaos by a fuzzy classifier, *Fuzzy Set and Systems*, *108*, 1–10, 1999.

Kalteh, A. M., P. Hjorth, and R. Berndtsson, Review of the self-organizing map (som) approach in water resources: Analysis, modelling and application, *Environmental Modelling & Software*, *23*, 2008.

Karunanithi, N., W. J. Grenney, D. Whitely, and K. Bovee, Neural networks for river flow prediction, *Journal of Computing in Civil Engineering*, *8*, 201–220, 1994.

Kastl, G. J., I. H. Fisher, and V. Jegatheesan, Evaluation of chlorine decay kinetics expressions for drinking water distribution systems modelling, *Journal of Water Science Research Technology - Aqua*, *48*, 219–226, 1999.

Kaufman, L., and P. J. Rousseeuw, *Finding groups in data. An introduction to cluster analysis.*, John Wiley & Sons, Brussels, 1990.

Kennard, R. W., and L. Stone, Computer aided design of experiments, *Technometrics*, *11*, 137–148, 1969.

Kingston, G. B., Bayesian artificial neural networks in water resources engineering, Ph.d., The University of Adelaide, 2006.

Kingston, G. B., H. Maier, and M. F. Lambert, Forecasting cyanobacteria with bayesian and deterministic artificial neural networks, in *IEEE World Congress on Computational Intelligence*, p. Proceedings on DVD, Vancouver, 2006a.

Kingston, G. B., H. R. Maier, and M. F. Lambert, A probabilistic method for assisting knowledge extraction from artificial neural networks used for hydrological prediction, *Mathematical and Computer Modelling*, *44*, 499–512, 2006b.

Knuth, D. E., *The Art of Computer Programming*, vol. 2, 3rd ed., Addison-Wesley, Boston, 1997.

Kohavi, R., and G. John, Wrappers for feature selection, *Artificial Intelligence*, *97*, 273–324, 1997.

Kohonen, T., *Self-organizing maps*, vol. 30 of *Springer Series in Information Sciences*, Springer-Verlag, Berlin, 1995.

Kollios, G., D. Gunopulos, N. Koudas, and S. Berchtold, Efficient biased sampling for approximate clustering and outlier detection in large data sets, *IEEE Transactions on Knowledge and Data Engineering*, *15*, 1170–1187, 2003.

Kpedekpo, G. M. K., Recent advances on some aspects of stratified sample design. a review of the literature, *Metrika*, *20*, 54–64, 1973.

Kwak, N., and C.-H. Choi, Input feature selection for classification problems, *IEEE Transactions on Neural Networks*, *13*, 143–159, 2002.

Laio, F., A. Porporato, R. Revelli, and L. Ridolfi, A comparison of nonlinear flood forecasting methods, *Water Resources Research*, *39*, 2003.

Lauzon, N., F. Anctil, and C. W. Baxter, Clustering of heterogeneous precipitation fields for the assessment and possible improvement of lumped neural network models for streamflow forecasts, *Hydrology and Earth System Sciences*, *10*, 485–494, 2006, lauzon, N. Anctil, F. Baxter, C. W.

LeBaron, B., and A. S. Weigend, A bootstrap evaluation of the effect of data splitting on financial time series, *IEEE Transactions on Neural Networks*, *9*, 213–220, 1998.

Lobbrecht, A. H., and D. P. Solomatine, Machine learning in real-time control of water systems, *Urban Water*, *4*, 283–289, 2002.

Lui, G. C. S., and W. K. Li, Modelling algal blooms using vector autoregressive model with exogenous variables and long memory filter, *Ecological Modelling*, *200*, 2007.

Machon, I., H. Lopez, J. Rodriguez-Iglesias, E. Maranon, and I. Vazquez, Simulation of a coke wastewater nitrification process using a feed-forward neuronal net, *Environmental Modelling and Software*, *22*, 1382–1387, 2007.

Maier, H., and G. C. Dandy, Determining inputs for neural network models of multivariate time series, *Microcomputers in Civil Engineering*, *12*, 353–368, 1997.

Maier, H., A. C. Zecchin, L. Radbone, and P. Goonan, Optimising the mutual information of ecological data clusters using evolutionary algorithms, *Mathematical and Computer Modelling*, *44*, 438–450, 2006.

Maier, H. R., Application of natural computing methods to water resources and environmental modelling, *Mathematical and Computer Modelling*, *44*, 413–414, 2006.

Maier, H. R., and G. C. Dandy, Neural network models for forecasting univariate time series, *Water Resources Research*, *32*, 1013–1022, 1996.

Maier, H. R., and G. C. Dandy, Understanding the behaviour and optimising the performance of back-propagation neural networks: an empirical study, *Environmental Modelling and Software*, *13*, 179–191, 1998.

Maier, H. R., and G. C. Dandy, Application of neural networks to forecasting of surface water quality variables: Issues, applications and challenges, *Environmental Modelling & Software*, *15*, 101–124, 2000.

Maier, H. R., N. Morgan, and C. W. K. Chow, Use of artificial neural networks for predicting optimal alum doses and treated water quality parameters, *Environmental Modelling and Software*, *19*, 485–494, 2004.

Mallows, C. L., Some comments on Cp, *Technometrics*, *15*, 661–675, 1973.

Marcoulides, G. A., and Z. Drezner, Model specification searches using ant colony optimization algorithms, *Structural Equation Modeling: A Multidisciplinary Journal*, *10*, 154–164, 2003.

May, R. J., G. C. Dandy, H. R. Maier, and T. M. K. G. Fernando, Critical values of a kernel-density based mutual information estimator, in *IEEE International Joint Conference on Neural Networks*, pp. 9997–10,002, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, 2006.

May, R. J., G. C. Dandy, H. R. Maier, and T. M. K. G. Fernando, Nonlinear variable selection for artificial neural networks using partial information, *Environmental Modelling and Software, 23,* 1312–1326, 2008a.

May, R. J., H. R. Maier, G. C. Dandy, and J. B. Nixon, Application of partial mutual information-based variable selection to ANN forecasting of water quality within water distribution systems, *Environmental Modelling and Software*, *23*, 1289–1299, 2008b.

May, R. J., H. R. Maier, and G. C. Dandy, Development of artificial neural networks for water quality modelling and analysis, in *Modelling of Pollutants in Complex Environmental Systems*, edited by G. Hanrahan, vol. 1, pp. 27–62, ILM Publications, London, UK, 2009a.

May, R. J., H. R. Maier, and G. C. Dandy, Data splitting for artificial neural networks using SOM-based stratified sampling, *Neural Networks*, *23*, 283–294, 2009b.

Miller, A. J., Selection of subsets of regression variables, *Journal of The Royal Statistical Society. Series A.*, *147*, 389–425, 1984.

Milot, J., M. J. Rodriguez, and J.-B. Srodes, Contribution of neural networks for modeling trihalomethanes occurrence in drinking water, *Journal of Water Resources Planning and Management, ASCE, 128*, 370–376, 2002.

Moddemeijer, R., A statistic to estimate the variance of the histogram-based mutual information estimator based on dependent pairs of observations, *Signal Processing*, *75*, 51–63, 1999.

Morari, M., and J. H. Lee, Model predictive control: Past, present and future, *Computers and chemical engineering*, pp. 667–682, 1999.

Mulvey, J. M., Multivarite stratified sampling by optimization, *Management Science*, *29*, 715–724, 1983.

Nanopoulos, A., Y. Theodoridis, and Y. Manolopoulos, An efficient and effective algorithm for density biased sampling, in *11th Conference on Information and Knowledge Management (CIKM)*, McLean, USA, 2002.

Narendra, K. S., Intelligent control, *IEEE Control Systems Magazine*, pp. 39–40, 1991.

Olsson, J., C. B. Uvo, K. Jinno, A. Kawamura, K. Nishiyama, N. Koreeda, T. Nakashima, and O. Morita, Neural networks for forecasting rainfall by atmospheric downscaling, *Journal of Hydraulic Engineering, ASCE*, *9*, 1–12, 2004.

Ozkaya, B., A. Demir, and M. Bilgili, Neural network prediction model for the methane fraction in biogas from field-scale landfill bioreactors, *Environmental Modelling and Software*, *22*, 815–822, 2007.

Palmer, C. R., and C. Faloutsos, Density biased sampling: An improved method for data mining and clustering, in *International conference on management of data (SIGMOD)*, Dalla, Texas, 2000.

Paninski, L., Estimation of entropy and mutual information, *Neural Computation*, *15*, 1191–1253, 2003.

Pearson, R. K., Outliers in process modeling and identification, *IEEE Transactions on Control Systems Technology*, *10*, 55–63, 2002.

Pires, J., F. Martins, S. Sousa, M. Alvim-Ferraz, and M. Pereira, Selection and validation of parameters in multiple linear and principal component regressions, *Environmental Modelling and Software*, *23*, 50–55, 2008.

Polycarpou, M. M., J. G. Uber, Z. Wang, F. Shang, and M. A. Brdys, Feedback control of water quality, *IEEE Control Systems Magazine*, *June*, 68–84, 2002.

Press, W. H., S. A. Tuekolsky, W. T. Vetterling, and B. P. Falnnery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, 1992.

Radcliffe, N. J., Genetic set recombination, in *Foundations of Genetic Algorithms 2*, edited by L. D. Whitley, p. 322, Morgan Kaufman, San Mateo, CA, 1993.

Raduly, B., K. V. Gernaey, A. G. Capodaglio, P. S. Mikkelsen, and M. Henze, Neural networks for rapid WWTP performance evaluation: Methodology and case study, *Environmental Modelling & Software*, *22*, 1208–1216, 2007.

Reeves, R. R., and S. J. Taylor, Selection of training data for neural networks by a genetic algorithm, in *Fifth International Conference on Parallel Problem Solving from Nature*, Amsterdam, 1998.

Rodriguez, M. J., and J.-B. Serodes, Neural network-based modelling of the adequate chlorine dosage for drinking water disinfection, *Canadian Journal of Civil Engineering*, *23*, 621–631, 1996.

Rodriguez, M. J., and J.-B. Serodes, Assessing empirical linear and non-linear modelling of residual chlorine in urban drinking water systems, *Environmental Modelling and Software*, *14*, 93–102, 1999.

Rodriguez, M. J., J.-B. Serodes, and P. Cote, Advanced chlorination control in drinking water systems using artificial neural networks, *Water Supply*, *15*, 159–168, 1997.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, Learning internal representations by error propagation, in *Parallel Distributed Processing: Explorations in the microstructure of cognition.*, edited by D. E. Rumelhard and J. L. McClelland, vol. 1: Foundations, MIT Press, Cambridge, MA, 1986.

Santos, P. J., A. G. Martins, and A. J. Pires, Designing the input vector to ann-based models for short-term load forecast in electricity distribution systems, *International Journal of Electrical Power and Energy Systems*, *29*, 338–47, 2005.

Sarle, W., Neural network FAQ. Periodic posting to the Usenet newsgroup comp.ai.neural-nets, 1997.

Schwarz, G., Estimating the dimension of a model, *Annals of Statistics*, *6*, 461–464, 1978.

Scott, D. W., *Multivariate density estimation: theory, practice and visualisation*, John Wiley and Sons, New York, 1992.

See, L., Data fusion methods for integrating data-driven hydrological models, in *Studies in Computational Intelligence*, vol. 79, pp. 1–18, Springer-Verlag, Berlin, 2008.

Serodes, J.-B., M. J. Rodriguez, and A. Ponton, Chlorcast©: a methodology for developing decision-making tools for chlorine disinfection control, *Environmental Modelling & Software*, *16*, 53–62, 2001.

Shahin, M., H. R. Maier, and M. B. Jaksa, Data division for developing neural networks applied to geotechnical engineering, *Journal of Computing in Civil Engineering*, *April*, 105–114, 2004.

Shanmuganathan, S., P. Sallis, and J. Buckeridge, Self-organising map methods in integrated modelling of environmental and economic systems, *Environmental Modelling and Software*, *21*, 1247–1256, 2006.

Shannon, C. E., A mathematical theory of communication, *Bell System Technical Journal*, *27*, 379–423, 1948.

Sharma, A., Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 - a strategy for system predictor identification, *Journal of Hydrology*, *239*, 232–239, 2000.

Shen, Q., J.-H. Jiang, J.-C. Tao, G.-L. Shen, and R.-Q. Yu, Modified ant colony optimization algorithm for variable selection in QSAR modeling : QSAR studies of cyclooxygenase inhibitors, *Journal of Chemical Information and Modeling*, *45*, 1024–1029, 2005.

Silverman, B. W., *Density estimation for statistics and data analysis*, Chapman and Hall, London, 1986.

Sindelar, R., and R. Babuska, Input selection for nonlinear regression models, *IEEE Transactions on Fuzzy Systems*, *12*, 688–696, 2004.

Sivakumar, B., A. W. Jayawardena, and T. M. K. G. Fernando, River flow forecasting: use of phase-space reconstruction and artificial neural networks approaches, *Journal of Hydrology*, *265*, 225–245, 2002.

Snee, R. D., Validation of regression models: Methods and examples, *Technometrics*, *19*, 415–428, 1977.

Soofi, E. S., and J. J. Retzer, Information importance of explanatory variables, in *IEE Conference in Honor of Arnold Zellner: Recent Developments in the Theory, Method and Application of Entropy Econometrics.*, Washington, 2003.

Sousa, S., F. Martins, M. Alvim-Ferraz, and M. Pereira, Multiple linear regression and artificial neural networks based on principle components to predict ozone concentrations, *Environmental Modelling and Software*, *22*, 97–103, 2007.

Specht, D. F., A general regression neural network, *IEEE Transactions on Neural Networks*, *2*, 568–576, 1991.

Sprevak, D., F. Azuaje, and H. Wang, A non-random data sampling method for classification model assessment, in *17th International Conference on Pattern Recognition (ICPR '04)*, vol. 3, pp. 406–409, Cambridge, UK, 2004.

Stephens, M. A., Edf statistics for goodness of fit and some comparisons, *Journal of the American Statistical Association*, *69*, 730–737, 1974.

Stephenson, N., *The Diamond Age,or A Young Lady's Illustrated Primer*, Bantam Books, New York, 1995.

Svozil, D., J. Pospichal, and V. Kvnasnicka, Neural network prediction of carbon-13 NMR chemical shifts of alkanes, *Journal of Chemical Information and Computer Sciences*, *35*, 924–928, 1995.

Tong, F., and X. Liu, Samples selection for artificial neural network training in preliminary structural design, *Tsinghua Science and Technology*, *10*, 233–239, 2005.

Torkkola, K., Feature extraction by non-parametric mutual information maximization, *Journal of Machine Learning Research*, *3*, 1415–1438, 2003.

Tourassi, G. A., E. D. Frederick, M. K. Markey, and C. E. Floyd Jr, Application of the mutual information criterion for feature selection in computer-aided diagnosis, *Medical Physics*, *28*, 2394–2402, 2001.

Tourassi, G. D., and C. E. Floyd, The effect of data sampling on the performance evaluation of artificial neural networks in medical diagnosis, *Medical Decision Making*, *17*, 186–192, 1997.

Trappenberg, T. P., J. Ouyang, and A. D. Back, Input variable selection: mutual information and linear mixing measures, *IEEE Transactions on Knowledge and Data Engineering*, *18*, 37–46, 2006.

Twomey, J. M., and A. E. Smith, Bias and variance of validation methods for function approximation neural networks under conditions of sparse data, *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, *28*, 417–430, 1998.

Van Zyl, J. E., J. Borthwick, and A. Hardy, OOTEN: An object-oriented programmers toolkit for epanet, in *International Conference on Advances in Water Supply Management*, London, UK, 2003.

Verstraeten, G., and D. Ven den Poel, Using predicted outcome stratified sampling to reduce the variability in predictive performance of a one-shot train-and-test split for individual customer predictions, *Tech. Rep. 360*, Ghent University, 2006.

Vesanto, J., SOM-based data visualization methods, *Intelligent Data Analysis*, *3*, 111–126, 1999.

Vesanto, J., and E. Alhoniemi, Clustering of the self-organizing map, *IEEE Transactions on Neural Networks*, *11*, 586–600, 2000.

Walski, T. M., D. V. Chase, D. A. Savic, W. Grayman, S. Beckwith, and E. Koelle, *Advanced water distribution modeling and management*, Haestad Methods, 1st ed., Haestad Press, Waterbury, 2003.

Wang, W., P. Van Gelder, J. K. Vrijling, and J. Ma, Forecasting daily streamflow using hybrid ANN models, *Journal of Hydrology*, *324*, 383–399, 2006.

Welk, A. L., A contribution towards real-time forecasting of algal blooms in drinking water reservoirs by means of artificial neural networks and evolutionary algorithms, PhD Thesis, University of Adelaide, 2008.

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

Wilcox, R. R., *Fundamentals of Modern Statistical Methods*, Springer-Verlag, New York, 2001.

Wold, H., Estimation of principal components and related models by iterative least squares, in *Multivariate Analysis*, edited by P. Krishnaiah, Academic Press, New York, 1966.

Wolpert, D. H., and D. R. Wolf, Estimating functions of probability distributions from a finite set of samples, *Physical Review E*, *52*, 6841–6854, 1995.

Wong, K. W., A neural fuzzy approach for well log and hydrocyclone data inerpretation, PhD Thesis, Curtin University of Technology, 1996.

Zhang, Q., Introduction to artificial neural networks in environmental modelling, *Journal of Environmental Engineering Science*, pp. i–iv, 2004.

Zhang, Q., S. J. Stanley, and D. W. Smith, Internal workings of feed-forward neural networks, *Journal of Environmental Engineering Science*, *3*, 1–12, 2004a.

Zhang, Q. J., A. A. Cudrak, R. Shariff, and S. J. Stanley, Implementing artificial neural network models for real-time water colour forecasting in a water treatment plant, *Journal of Environmental Engineering Science*, *3*, 15–23, 2004b.

Zhou, Z.-H., J. Wu, and W. Tang, Ensembling neural networks: Many could be better than all, *Artificial Intelligence*, *137*, 239–263, 2002.

Appendix A

Critical Values of *I* and *R*

This appendix presents complete tables of the critical value of mutual information, I^*, and the Pearson correlation, R^*, which form the basis for termination criteria used for PMIS and PCIS in Chapter 5.

Tables A.1—A.6 contain estimates of the critical value of the KDE mutual information estimator described in Section 5.3. The estimates were obtained by Monte Carlo simulation of the MI for independently sampled (uncorrelated) Gaussian noise variables. Estimates were obtained for $I(X;Y)$ for a range of dimensions of both X and Y.

Critical values for the Pearson correlation (Table A.7) were computed using the analytical method based on the assumption of a t-distribution of the error for a sample estimate with degrees of freedom, $d_f = n - 2$. Critical values of R are determined according to the relationship

$$R^* = \sqrt{\frac{t_c^2}{d_f + t_c^2}}.$$

Table A.1: Critical values of the KDE estimate of I obtained by Monte Carlo simulation for the bivariate (two-dimensional) case $I(x_1; y)$.

	Confidence level, α			
n	0.5	0.1	0.05	0.01
50	0.1276	0.1990	0.2224	0.2705
60	0.1195	0.1825	0.2031	0.2452
70	0.1131	0.1694	0.1879	0.2254
80	0.1076	0.1592	0.1756	0.2091
90	0.1029	0.1506	0.1657	0.1973
100	0.0987	0.1429	0.1572	0.1858
120	0.0920	0.1309	0.1434	0.1688
140	0.0864	0.1211	0.1321	0.1546
160	0.0821	0.1138	0.1237	0.1444
180	0.0783	0.1072	0.1166	0.1356
200	0.0750	0.1019	0.1103	0.1276
220	0.0722	0.0975	0.1055	0.1215
240	0.0695	0.0932	0.1005	0.1158
260	0.0671	0.0894	0.0965	0.1108
280	0.0652	0.0862	0.0928	0.1062
300	0.0633	0.0834	0.0896	0.1022
400	0.0559	0.0724	0.0775	0.0876
500	0.0507	0.0646	0.0689	0.0775
600	0.0468	0.0589	0.0627	0.0702
700	0.0436	0.0544	0.0578	0.0644
800	0.0411	0.0509	0.0539	0.0597
900	0.0389	0.0479	0.0507	0.0563
1000	0.0372	0.0455	0.0481	0.0531
2000	0.0269	0.0318	0.0333	0.0361
3000	0.0221	0.0257	0.0268	0.0289
4000	0.0192	0.0221	0.0230	0.0247
5000	0.0172	0.0196	0.0204	0.0218

Table A.2: Critical values of the KDE estimate of I obtained by Monte Carlo simulation for the multivariate (three-dimensional) case $I(x_1, x_2; y)$.

	Confidence level, α			
n	0.5	0.1	0.05	0.01
50	0.3342	0.4443	0.4774	0.5418
60	0.3200	0.4183	0.4479	0.5046
70	0.3078	0.3976	0.4251	0.4757
80	0.2975	0.3797	0.4044	0.4515
90	0.2885	0.3648	0.3877	0.4333
100	0.2805	0.3514	0.3721	0.4136
120	0.2665	0.3299	0.3490	0.3854
140	0.2551	0.3123	0.3296	0.3627
160	0.2458	0.2984	0.3136	0.3436
180	0.2371	0.2858	0.3000	0.3281
200	0.2299	0.2751	0.2884	0.3142
220	0.2236	0.2662	0.2790	0.3032
240	0.2177	0.2580	0.2701	0.2934
260	0.2125	0.2504	0.2618	0.2839
280	0.2077	0.2436	0.2543	0.2751
300	0.2032	0.2377	0.2478	0.2674
400	0.1856	0.2140	0.2225	0.2388
500	0.1727	0.1971	0.2043	0.2183
600	0.1627	0.1843	0.1906	0.2030
700	0.1545	0.1738	0.1795	0.1905
800	0.1479	0.1655	0.1707	0.1809
900	0.1420	0.1582	0.1631	0.1723
1000	0.1370	0.1522	0.1567	0.1651
2000	0.1075	0.1168	0.1195	0.1248
3000	0.0928	0.0999	0.1019	0.1058
4000	0.0835	0.0892	0.0909	0.0941
5000	0.0768	0.0817	0.0831	0.0859

Table A.3: Critical values of the KDE estimate of I obtained by Monte Carlo simulation for the multivariate (four-dimensional) case $I(x_1, x_2, x_3; y)$.

	Confidence level, α			
n	0.5	0.1	0.05	0.01
50	0.5544	0.6922	0.7313	0.8061
60	0.5411	0.6658	0.7019	0.7696
70	0.5286	0.6429	0.6768	0.7391
80	0.5182	0.6243	0.6546	0.7125
90	0.5087	0.6075	0.6362	0.6902
100	0.4994	0.5929	0.6197	0.6720
120	0.4839	0.5683	0.5926	0.6383
140	0.4705	0.5471	0.5692	0.6105
160	0.4594	0.5301	0.5506	0.5885
180	0.4483	0.5148	0.5345	0.5711
200	0.4395	0.5012	0.5191	0.5531
220	0.4309	0.4898	0.5070	0.5390
240	0.4238	0.4793	0.4953	0.5252
260	0.4165	0.4695	0.4844	0.5136
280	0.4104	0.4605	0.4752	0.5031
300	0.4043	0.4528	0.4667	0.4928
400	0.3800	0.4205	0.4323	0.4543
500	0.3614	0.3968	0.4071	0.4264
600	0.3466	0.3781	0.3871	0.4043
700	0.3343	0.3627	0.3711	0.3867
800	0.3239	0.3502	0.3577	0.3722
900	0.3149	0.3390	0.3459	0.3592
1000	0.3071	0.3297	0.3362	0.3487
2000	0.2578	0.2723	0.2765	0.2844
3000	0.2319	0.2430	0.2463	0.2524
4000	0.2146	0.2238	0.2264	0.2314
5000	0.2019	0.2098	0.2122	0.2165

Table A.4: Critical values of the KDE estimate of I obtained by Monte Carlo simulation for the multivariate (four-dimensional) case $I(x_1, x_2; y_1, y_2)$.

	Confidence level, α			
n	0.5	0.1	0.05	0.01
50	0.7585	0.9288	0.9786	1.0692
60	0.7401	0.8946	0.9388	1.0228
70	0.7221	0.8643	0.9051	0.9803
80	0.7066	0.8381	0.8758	0.9467
90	0.6933	0.8161	0.8515	0.9189
100	0.6797	0.7952	0.8283	0.8928
120	0.6574	0.7616	0.7916	0.8480
140	0.6385	0.7330	0.7601	0.8120
160	0.6222	0.7099	0.7352	0.7835
180	0.6071	0.6887	0.7123	0.7583
200	0.5942	0.6705	0.6925	0.7334
220	0.5824	0.6544	0.6752	0.7136
240	0.5717	0.6399	0.6599	0.6970
260	0.5617	0.6265	0.6453	0.6809
280	0.5526	0.6142	0.6323	0.6665
300	0.5440	0.6037	0.6205	0.6536
400	0.5094	0.5591	0.5734	0.6004
500	0.4833	0.5267	0.5388	0.5632
600	0.4623	0.5009	0.5121	0.5335
700	0.4450	0.4799	0.4900	0.5088
800	0.4306	0.4627	0.4721	0.4895
900	0.4179	0.4473	0.4556	0.4714
1000	0.4067	0.4345	0.4426	0.4575
2000	0.3384	0.3559	0.3610	0.3706
3000	0.3026	0.3160	0.3199	0.3273
4000	0.2788	0.2899	0.2930	0.2991
5000	0.2615	0.2711	0.2738	0.2790

Table A.5: Critical values of the KDE estimate of I obtained by Monte Carlo simulation for the multivariate (five-dimensional) case $I(x_1, x_2, x_3, x_4; y)$.

	Confidence level, α			
n	0.5	0.1	0.05	0.01
50	0.7301	0.8818	0.9238	1.0015
60	0.7225	0.8608	0.8997	0.9730
70	0.7154	0.8438	0.8794	0.9480
80	0.7083	0.8285	0.8616	0.9240
90	0.7019	0.8142	0.8462	0.9049
100	0.6954	0.8022	0.8325	0.8902
120	0.6835	0.7808	0.8086	0.8600
140	0.6725	0.7626	0.7878	0.8354
160	0.6642	0.7469	0.7706	0.8163
180	0.6548	0.7335	0.7561	0.7992
200	0.6469	0.7211	0.7421	0.7808
220	0.6396	0.7102	0.7299	0.7682
240	0.6335	0.6999	0.7188	0.7548
260	0.6266	0.6906	0.7088	0.7430
280	0.6208	0.6823	0.6998	0.7333
300	0.6153	0.6745	0.6913	0.7226
400	0.5920	0.6420	0.6566	0.6835
500	0.5732	0.6179	0.6303	0.6541
600	0.5581	0.5980	0.6092	0.6313
700	0.5448	0.5813	0.5919	0.6110
800	0.5336	0.5674	0.5769	0.5949
900	0.5237	0.5550	0.5638	0.5811
1000	0.5150	0.5445	0.5531	0.5692
2000	0.4572	0.4769	0.4825	0.4931
3000	0.4248	0.4402	0.4445	0.4529
4000	0.4023	0.4152	0.4188	0.4257
5000	0.3854	0.3966	0.3998	0.4058

Table A.6: Critical values of the KDE estimate of I obtained by Monte Carlo simulation for the multivariate (five-dimensional) case $I(x_1, x_2, x_3; y_1, y_2)$.

	Confidence level, α			
n	0.5	0.1	0.05	0.01
50	1.1534	1.3568	1.4137	1.5210
60	1.1403	1.3292	1.3808	1.4821
70	1.1283	1.3022	1.3517	1.4413
80	1.1158	1.2780	1.3239	1.4084
90	1.1056	1.2588	1.3022	1.3825
100	1.0939	1.2394	1.2809	1.3623
120	1.0738	1.2063	1.2444	1.3155
140	1.0551	1.1782	1.2132	1.2776
160	1.0401	1.1540	1.1877	1.2474
180	1.0243	1.1317	1.1632	1.2207
200	1.0108	1.1116	1.1413	1.1953
220	0.9974	1.0947	1.1222	1.1739
240	0.9869	1.0780	1.1038	1.1538
260	0.9752	1.0632	1.0884	1.1347
280	0.9653	1.0501	1.0744	1.1201
300	0.9556	1.0370	1.0603	1.1039
400	0.9154	0.9841	1.0037	1.0410
500	0.8834	0.9445	0.9617	0.9945
600	0.8574	0.9120	0.9276	0.9577
700	0.8352	0.8851	0.8993	0.9261
800	0.8161	0.8625	0.8756	0.9005
900	0.7993	0.8422	0.8542	0.8775
1000	0.7848	0.8250	0.8364	0.8578
2000	0.6881	0.7148	0.7223	0.7368
3000	0.6345	0.6552	0.6612	0.6723
4000	0.5977	0.6150	0.6200	0.6296
5000	0.5701	0.5850	0.5894	0.5973

Table A.7: Critical values of the Pearson coefficient of cross-correlation for the bivariate case, $R(x, y)$.

	Confidence level, α			
n	0.5	0.10	0.05	0.01
50	0.0095	0.0775	0.1001	0.1523
60	0.0079	0.0645	0.0835	0.1278
70	0.0067	0.0553	0.0716	0.1100
80	0.0059	0.0483	0.0627	0.0965
90	0.0052	0.0429	0.0558	0.0860
100	0.0047	0.0386	0.0502	0.0776
120	0.0039	0.0322	0.0418	0.0648
140	0.0033	0.0275	0.0359	0.0557
160	0.0029	0.0241	0.0314	0.0488
180	0.0026	0.0214	0.0279	0.0434
200	0.0023	0.0193	0.0251	0.0391
220	0.0021	0.0175	0.0228	0.0356
240	0.0019	0.0160	0.0209	0.0326
260	0.0018	0.0148	0.0193	0.0301
280	0.0016	0.0137	0.0179	0.0280
300	0.0015	0.0128	0.0167	0.0261
400	0.0011	0.0096	0.0126	0.0196
500	0.0009	0.0077	0.0100	0.0157
600	0.0008	0.0064	0.0084	0.0131
700	0.0007	0.0055	0.0072	0.0112
800	0.0006	0.0048	0.0063	0.0098
900	0.0005	0.0043	0.0056	0.0087
1 000	0.0005	0.0038	0.0050	0.0079
2 000	0.0002	0.0019	0.0025	0.0039
3 000	0.0002	0.0013	0.0017	0.0026
4 000	0.0001	0.0010	0.0013	0.0020
5 000	0.0001	0.0008	0.0010	0.0016

Appendix B

IVS Performance Data

This appendix provides full tables of results for the application of PMIS and PCIS to seven benchmark synthetic datasets using different termination criteria, which was summarised in Chapter 5.

Table B.1: Percentage correct model specifications for PMIS and PCIS (50-sample datasets)

Algorithm	Criterion	Linear Models						Non-linear Models														
		AR4			AR9			TAR2			TAR1			Friedman			Friedman (15)			Mexican-Hat		
		f_u^*	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o
PCIS	A	30	40	30	40	30	30	67	23	10	77	10	13	87	13	0	93	3	3	100	0	0
	B	53	33	13	63	37	0	87	13	0	97	0	3	100	0	0	97	3	0	100	0	0
	C	40	30	30	47	27	27	70	23	7	80	10	10	97	3	0	97	0	3	100	0	0
	D	53	40	7	63	37	0	87	13	0	100	0	0	100	0	0	100	0	0	100	0	0
	E	43	20	37	47	27	27	70	23	7	80	10	10	100	0	0	100	0	0	100	0	0
	F	53	40	7	73	27	0	87	13	0	100	0	0	100	0	0	100	0	0	100	0	0
	G	17	10	73	17	23	60	30	17	53	57	7	37	83	17	0	70	10	20	100	0	0
PMIS	A	40	13	47	40	23	37	23	30	47	73	17	10	40	47	13	40	30	30	17	40	43
	B	50	17	33	50	33	17	53	23	23	93	7	0	60	33	7	60	30	10	50	33	17
	C	40	10	50	20	7	73	27	33	40	77	13	10	47	37	17	40	27	33	17	40	43
	D	53	27	20	23	7	70	63	17	20	93	7	0	73	23	3	67	27	7	60	30	10
	E	90	10	0	100	0	0	100	0	0	100	0	0	100	0	0	100	0	0	100	0	0
	F	100	0	0	100	0	0	100	0	0	100	0	0	100	0	0	100	0	0	100	0	0
	G	57	30	13	93	7	0	77	23	0	80	20	0	93	7	0	90	10	0	20	80	0
	H	83	7	10	87	10	3	73	20	7	83	10	7	100	0	0	83	13	3	77	23	0

*f_u = frequency under-specified, f_c = frequency correctly specified, f_o = frequency over-specified

Table B.2: Percentage correct model specifications for PMIS and PCIS (100-sample datasets)

Algorithm	Criterion	Linear Models						Non-linear Models														
		AR4			AR9			TAR2			TAR1			Friedman			Friedman (15)			Mexican-Hat		
		f_u^*	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o
PCIS	A	0	60	40	7	53	40	80	0	20	97	0	3	90	10	0	100	0	0	100	0	0
	B	0	83	17	17	73	10	87	7	7	100	0	0	90	10	0	100	0	0	100	0	0
	C	0	67	33	7	50	43	77	3	20	97	0	3	90	10	0	100	0	0	100	0	0
	D	0	93	7	20	73	7	93	7	0	100	0	0	90	10	0	100	0	0	100	0	0
	E	0	60	40	7	53	40	77	3	20	97	3	0	90	10	0	100	0	0	100	0	0
	F	0	93	7	20	73	7	93	7	0	100	0	0	90	10	0	100	0	0	100	0	0
	G	0	23	77	7	10	83	50	3	47	87	0	13	77	7	17	80	7	13	100	0	0
PMIS	A	0	23	77	23	23	53	3	7	90	50	37	13	0	70	30	0	60	40	3	57	40
	B	10	53	37	27	40	33	7	37	57	70	27	3	0	83	17	0	73	27	3	77	20
	C	0	33	67	13	3	83	3	10	87	43	40	17	0	77	23	0	63	37	3	67	30
	D	7	63	30	20	7	73	7	53	40	77	20	3	0	90	10	0	83	17	3	83	13
	E	77	23	0	97	3	0	90	10	0	100	0	0	100	0	0	100	0	0	100	0	0
	F	93	7	0	100	0	0	97	3	0	100	0	0	100	0	0	100	0	0	100	0	0
	G	13	60	27	77	10	13	60	30	10	47	53	0	73	27	0	47	53	0	0	97	3
	H	70	17	13	67	30	3	30	43	27	63	33	3	93	7	0	30	60	10	13	70	17

*f_u = frequency under-specified, f_c = frequency correctly specified, f_o = frequency over-specified

Table B.3: Percentage correct model specifications for PMIS and PCIS (500-sample datasets)

Algorithm	Criterion	Linear Models						Non-linear Models														
		AR4			AR9			TAR2			TAR1			Friedman			Friedman (15)			Mexican-Hat		
		f_u^*	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o
PCIS	A	0	60	40	0	53	47	0	23	77	93	0	7	97	0	3	100	0	0	100	0	0
	B	0	93	7	0	67	33	0	50	50	97	3	0	97	0	3	100	0	0	100	0	0
	C	0	63	37	0	50	50	0	20	80	93	3	3	97	0	3	100	0	0	100	0	0
	D	0	93	7	0	73	27	3	67	30	97	3	0	100	0	0	100	0	0	100	0	0
	E	0	70	30	0	47	53	0	23	77	97	0	3	97	0	3	100	0	0	100	0	0
	F	0	97	3	0	80	20	3	70	27	97	3	0	100	0	0	100	0	0	100	0	0
	G	0	7	93	0	3	97	0	0	100	80	3	17	90	3	7	90	0	10	100	0	0
PMIS	A	0	53	47	0	23	77	0	7	93	0	60	40	0	80	20	0	60	40	0	47	53
	B	0	70	30	0	63	37	0	20	80	0	83	17	0	93	7	0	83	17	0	87	13
	C	0	60	40	0	7	93	0	3	97	0	67	33	0	83	17	0	67	33	0	50	50
	D	0	80	20	0	17	83	0	23	77	0	90	10	0	100	0	0	87	13	0	87	13
	E	0	100	0	10	90	0	0	100	0	93	7	0	0	100	0	0	100	0	0	100	0
	F	0	100	0	30	70	0	0	100	0	100	0	0	0	100	0	0	100	0	0	100	0
	G	0	97	3	67	33	0	27	67	7	0	100	0	0	100	0	0	100	0	0	100	0
	H	10	73	17	3	90	7	0	67	33	0	100	0	77	17	7	0	93	7	0	87	13

*f_u = frequency under-specified, f_c = frequency correctly specified, f_o = frequency over-specified

Table B.4: Percentage correct model specifications for PMIS and PCIS (1000-sample datasets)

Algorithm	Criterion	Linear Models						Non-linear Models														
		AR4			AR9			TAR2			TAR1			Friedman			Friedman (15)			Mexican-Hat		
		f_u^*	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o	f_u	f_c	f_o
PCIS	A	0	53	47	0	63	37	0	40	60	90	0	10	93	7	0	97	3	0	100	0	0
	B	0	83	17	0	90	10	0	63	37	93	3	3	100	0	0	100	0	0	100	0	0
	C	0	63	37	0	63	37	0	40	60	93	0	7	93	7	0	97	3	0	100	0	0
	D	0	90	10	0	90	10	0	73	27	97	0	3	100	0	0	100	0	0	100	0	0
	E	0	57	43	0	63	37	0	40	60	93	0	7	90	10	0	97	3	0	100	0	0
	F	0	93	7	0	90	10	0	73	27	97	0	3	100	0	0	100	0	0	100	0	0
	G	0	23	77	0	17	83	0	13	87	80	3	17	80	7	13	83	3	13	100	0	0
PMIS	A	0	27	73	0	20	80	0	0	100	0	53	47	0	63	37	0	63	37	0	37	63
	B	0	63	37	0	43	57	0	3	97	0	87	13	0	87	13	0	80	20	0	83	17
	C	0	33	67	0	10	90	0	0	100	0	50	50	0	70	30	0	63	37	0	40	60
	D	0	70	30	0	17	83	0	7	93	0	93	7	0	87	13	0	90	10	0	90	10
	E	0	100	0	0	100	0	0	100	0	7	93	0	0	100	0	0	100	0	0	100	0
	F	0	100	0	0	100	0	0	100	0	23	77	0	0	100	0	0	100	0	0	100	0
	G	0	100	0	67	33	0	10	83	7	0	80	20	0	100	0	0	100	0	0	100	0
	H	3	87	10	0	90	10	0	50	50	0	97	3	90	3	7	0	93	7	0	70	30

*f_u = frequency under-specified, f_c = frequency correctly specified, f_o = frequency over-specified