

Short-term stream flow forecasting and downstream gap infilling using machine learning techniques

by

Melise Steyn



UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science (Applied Mathematics) in the Faculty of Science at Stellenbosch University

1918-2018

Supervisor: Prof. G.J.F. Smit
Co-supervisors: Dr J.M. Wilms
Dr W.H. Brink

March 2018

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2018

Copyright © 2018 Stellenbosch University
All rights reserved.

Abstract

Stream flow is an important component in the hydrological cycle and plays a vital role in many hydrological applications. Accurate stream flow forecasts may be used for the study of various hydro-environmental aspects and may assist in reducing the consequences of floods. The utility of time series records for stream flow analyses is often dependent on continuous, uninterrupted observations. However, interruptions are often unavoidable and may negatively impact the sustainable management of water resources. This study proposes the application of machine learning techniques to address these hydrological challenges.

The first part of this study focuses on single station short-term stream flow forecasting for river basins where historical time series data are available. Two machine learning techniques were investigated, namely support vector regression and multilayer perceptrons. Each model was trained on historical stream flow and precipitation data to forecast stream flow with a lead time of up to seven days. The Shoalhaven, Herbert and Adelaide rivers in Australia were considered for experimentation. The predictive performance of each model was determined by the Pearson correlation coefficient, the root mean squared error and the Nash-Sutcliffe efficiency, and the predictive capabilities of the models were compared to that of a physically based stream flow forecasting model currently supplied by the Australian Bureau of Meteorology. Based on the results, it was concluded that the machine learning models have the ability to overcome certain challenges faced by physically based models and the potential to be useful stream flow forecasting tools in river basin modelling.

The second part of this study investigates the ability of support vector regression and multilayer perceptron models to infill incomplete stream flow records. The infilling techniques relied upon data from donor stations and rain gauges within close proximity to the station considered for infilling. A case study was conducted on a channel in the Goulburn basin in Australia. The results showed the promising role of machine learning applications for the infilling of gaps in stream flow records and indicated that data from donor stations contribute more to the success of these models compared to precipitation data.

Uittreksel

Stroomvloei is 'n belangrike komponent in die hidrologiese siklus en speel 'n prominente rol in verskeie hidrologiese toepassings. Akkurate stroomvloei-voorspellings kan vir die bestudering van verskeie hidrologiese omgewingsaspekte gebruik word en kan help om die nagevolge van vloede te verminder. Die gebruik van tydreksdata vir stroomvloei-analise is dikwels afhanklik van ononderbroke waarnemings. Onderbrekings is egter dikwels onvermydelik en kan 'n negatiewe impak op die volhoubare bestuur van waterhulpbronne hê. In hierdie studie is die toepassing van masjienleertegnieke met die doel om hierdie hidrologiese uitdagings aan te spreek, bestudeer.

In die eerste gedeelte van hierdie studie is daar op korttermyn stroomvloei-voorspellings by meetstasies wat oor beskikbare historiese tydreksdata beskik, gefokus. Twee masjienleertegnieke is ondersoek, naamlik steunvektor-regressie en multi-laag perseptron modelle. Elke model is op historiese stroomvloei- en reënvaldata afgerig om stroomvloei tot en met sewe dae vooruit te voorspel. Eksperimente is op die Shoalhaven, Herbert en Adelaide riviere in Australië uitgevoer. Die voorspellingsvermoëns van elke model is deur die Pearson-korrelasiekoëffisiënt, die wortel-gemiddelde-kwadraat fout en die Nash-Sutcliffe-doeltreffendheid bepaal, en is met dié van 'n fisiese stroomvloei-voorspellingsmodel wat tans deur die Australiese Buro vir Meteorologie verskaf word, vergelyk. Op grond van die resultate is daar tot die gevolgtrekking gekom dat die masjienleermodelle oor die vermoëns beskik om sekere uitdagings waarmee fisiese modelle gekonfronteer word, te oorkom, en dat hulle 'n waardevolle bydrae tot die modellering van riverkomme kan lewer.

In die tweede gedeelte van hierdie studie is steunvektor-regressie en multi-laag perseptron modelle se vermoëns om onvolledige stroomvloei-state te vul, ondersoek. Die invul-tegnieke was afhanklik van data vanaf ander nabygeleë meetstasies en reënmeters. 'n Gevallestudie is op 'n kanaal in die Goulburn opvangsgebied in Australië uitgevoer. Die resultate het die belowende rol van masjienleertoepassings op die invul van gapings in stroomvloei-state getoon en aangedui dat data van meetstasies 'n groter bydrae tot die sukses van hierdie modelle lewer in vergelyking met reënvaldata.

Acknowledgements

All glory be to God Almighty, my Heavenly Father, Saviour and Ultimate Teacher for making the completion of this study a reality. What a privilege to serve and worship a God who answers my prayers, renews my strength, gives me insight and loves me unconditionally. I will forever be thankful for all the opportunities that He has blessed me with. “God arms me with strength, and he makes my way perfect. He makes me as surefooted as a deer, enabling me to stand on mountain heights. You have given me your shield of victory. Your right hand supports me; your help has made me great.” Psalm 18:32-33,35

I would also like to express my sincere appreciation to the following individuals and institutions who contributed to the completion of this research:

The Modelling and Digital Science (MDS) unit of the Council for Scientific and Industrial Research (CSIR), for funding my studies since 2012. They made my dream of studying at Stellenbosch University a reality, and also gave me the financial support to present my work at a conference in Spain earlier this year. A special thank you to my managers, Dr Onno Ubbink and Dr Kobie Smit, for always going the extra mile for the MDS group, and to my colleagues at the CSIR for their constant support and encouragement the past two years.

My supervisors, Prof. Francois Smit and Dr Willie Brink at Stellenbosch University, and Dr Josefine Wilms at the CSIR, for providing invaluable guidance and assistance. Their critical appraisals, suggestions and keen eye for detail have contributed greatly to the completion of this study.

My beloved parents, Francois and Sharon du Toit, and brother, Francois, for their endless love, support, prayers and encouragement. Words cannot describe how thankful I am for the tremendous sacrifices that my parents have made to ensure an excellent education and endless opportunities. I would also like to express my gratitude to my parents-in-law and friends who have never ceased to pray for me.

My biggest support, my cheerleader, my best friend, my greatest love, my husband: Wim Steyn. His infectious excitement about life, his purpose-driven lifestyle and his unfailing love, encouragement and support have made these past two years not only bearable, but also memorable. Thank you for being the best husband any wife could possibly have.

Nomenclature

Variables

a_i	Output of node i in particular MLP layer
b	Bias in MLP formulation
c	Variable in SVR formulation
C	Penalty parameter in SVR formulation
d	Forecasting lead time (days)
D	Stream flow at downstream station (ML/day)
ϵ	Margin of error in SVR formulation
f_i	Forecasted output variable
\bar{f}	Mean forecasted output variable
h	Number of hidden nodes in an MLP model
K	Number of folds considered for cross-validation
l	Lag time (days)
m	Number of output variables in the test set
n	Number of training samples
N	Number of input nodes in an MLP model
NSE	Nash-Sutcliffe efficiency
p	Number of preceding precipitation values in the input vector of the machine learning model
P	Precipitation (mm)
q	Number of preceding stream flow values in the input vector of the machine learning model

Q	Stream flow (ML/day)
r	Pearson's correlation coefficient
RMSE	Root mean squared error
s	Step size in MLP optimisation algorithm
t	Current day
u	Number of preceding stream flow values from upstream station in the input vector of a machine learning model
U	Stream flow at upstream station
w_{ij}	Weight connecting layer i with layer j
x_i	Input variable
\bar{x}	Mean input variable
X	Original input space
y_i, y	True output variable
\bar{y}	Mean true output variable
z	Weighted sum of a specific MLP node's input
$\alpha, \alpha^*, \eta, \eta^*$	Lagrange multipliers
γ, r, v	Kernel-specific hyperparameters in SVR
λ	Dual variable
μ	Mean value of a dataset
ξ, ξ^*	Slack variables in SVR formulation
σ	Standard deviation of a dataset
$\Phi(X)$	Feature space

Vectors

d	Direction of descent
f	Forecasted output vector
Q	Stream flow vector (ML/day)

\mathbf{w}	Weight vector
\mathbf{x}	Input vector
\mathbf{y}	True output vector

Matrices

\mathbf{H}	Hessian matrix
--------------	----------------

Functions

E	Error function
f_{loss}	ϵ -insensitive loss function
$f(\mathbf{x})$	Target function
g	Activation function
k	Kernel function
L_D	Dual Lagrangian function
L_P	Primal Lagrangian function

Abbreviations

BOM	Bureau of Meteorology
CDO	Climate Data Online
HRS	Hydrologic Reference stations
IDW	Inverse distance weighting
L-BFGS	Limited-memory Broyden-Fletcher-Goldfarb-Shanno
MLP	Multilayer perceptron
RBF	Radial basis function
SVM	Support vector machine
SVR	Support vector regression

Subscripts

i, j, k	Integer index
-----------	---------------

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives and domain of this study	3
1.3	Thesis layout	4
1.4	Publications from this study	5
2	Stream flow and hydrographs	6
2.1	Hydrograph shape	6
2.1.1	Rising limb	7
2.1.2	Crest segment	8
2.1.3	Recession limb	8
2.2	Ephemeral, intermittent and perennial rivers	8
2.3	Factors affecting a hydrograph	9
2.3.1	Climatic factors	10
2.3.2	Topographic and geologic factors	12
3	Data-driven modelling	15
3.1	Fundamentals of machine learning	15
3.1.1	Training, validation and testing	16
3.1.2	The bias-variance trade-off	18
3.1.3	Preparation of data	19
3.1.4	Performance evaluation	20
3.2	Machine learning techniques in hydrology	22
3.3	Support vector regression	22
3.3.1	Model formulation	23
3.3.2	Nonlinearity and kernels	27
3.3.3	Advantages and drawbacks	28
3.4	Neural networks	29
3.4.1	Model formulation	29
3.4.2	Network architecture	31
3.4.3	Network training	32
3.4.4	Advantages and drawbacks	34
4	Single station forecasting	35

4.1	Methodology	36
4.1.1	Study area and data	36
4.1.2	Selection of input variables	40
4.1.3	Preprocessing	42
4.1.4	SVR hyperparameters	42
4.1.5	MLP architecture	43
4.1.6	Software	43
4.2	Results	44
4.2.1	Parameter selection	44
4.2.2	Performance evaluation	44
4.3	Discussion	51
5	Gap infilling of stream flow records	56
5.1	Infilling techniques	56
5.2	Methodology	58
5.2.1	Study area and data	58
5.2.2	Selection of input variables	60
5.2.3	Preprocessing	62
5.2.4	Hyperparameters and network architecture	62
5.3	Results	62
5.3.1	Feature selection	62
5.3.2	Performance evaluation	65
5.4	Discussion	69
6	Conclusions and recommendations	71
	References	73

Chapter 1

Introduction

1.1 Motivation

Stream flow is an important component in the hydrological cycle and plays a vital role in many hydraulic and hydrological applications. Research on model-generated stream flow is used by river engineers and scientists for the study of various hydro-environmental aspects, such as the increasing international concern of riverine pollution and the growing flood stages of rivers (Falconer *et al.*, 2005). Consequences of natural disasters, such as floods, can be lessened or even prevented through accurate stream flow forecasts (Raghavendra and Deka, 2014).

Modern river basin management, based on the prediction of stream flow and the analysis of different environmental scenarios, is reliant on the adequacy of the particular hydrological model used (Falconer *et al.*, 2005; Solomatine and Ostfeld, 2008). A popular conventional model for stream flow forecasting is a physically based rainfall-runoff model. This model is used to transform rainfall estimations to runoff, which in turn may be used to determine stream flow by modelling the hydrologic processes within a catchment. As illustrated in Figure 1.1, these processes typically include interception, infiltration, evaporation, snowmelt, retention and detention storages, soil water movement, percolation to ground water, overland flow, open channel flow and subsurface flow (Knapp *et al.*, 1991). According to Perrin *et al.* (2003), it can be challenging to choose an appropriate model structure and complexity for accurate simulation of hydrological behaviour at catchment scale. If the model is too simple, it might prevent sufficient flexibility for an adequate representation of hydrological events within the catchment, whereas a model that is too complex may result in model robustness problems (Perrin *et al.*, 2003). These challenges might limit the modelling accuracy of a physically based model.

During the past decade, major progress has been made in the study of data-driven models to simulate hydrological processes within a catchment (Solo-

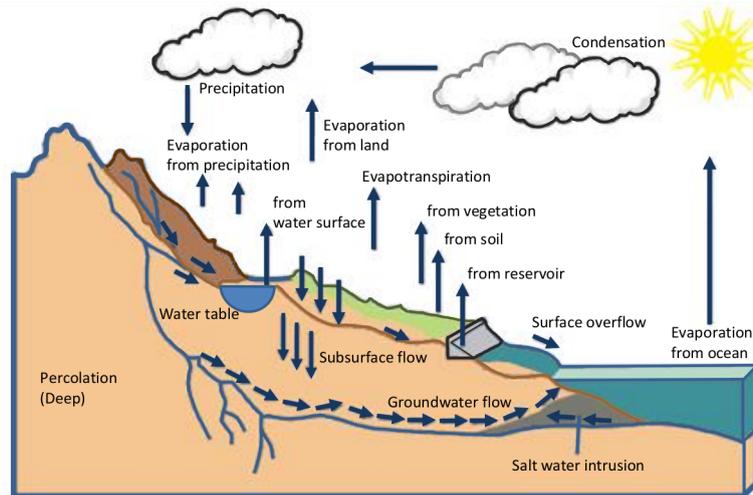


Figure 1.1: An illustration of the hydrological processes that have to be taken into account when modelling stream flow with a physically based rainfall-runoff model, redrawn from Encyclopaedia Britannica (2017).

matine and Ostfeld, 2008). Data-driven models are based on observed data that characterise the system under study. While physically based models involve equations derived from physical processes within the specific system, data-driven models include equations obtained from analysing time series data (Solomatine and Ostfeld, 2008). Various processes within a river basin are characterised by measurable state variables, such as stream flow, precipitation, temperature and humidity. A river basin for which time series records are available may therefore be a good candidate for the implementation of data-driven models.

The utility of time series records for stream flow analyses is often dependent on continuous, uninterrupted observations. The production and management of hydrometric data over a long period of time is, however, a challenging task. Technical or maintenance problems of a gauging station may affect its ability to generate flow measurements and may result in an incomplete dataset. Factors responsible for discontinuities in available records include the malfunctioning of equipment, flood damages, infrequent calibration of sensors and upgrades to existing equipment for more sophisticated measuring techniques. Gaps in a time series record indicate a loss of information and, according to Tencaliec *et al.* (2015), may lead to inaccurate and unreliable hydrological analyses. Incomplete datasets increase the complexity and uncertainty of hydrological modelling, and even very small gaps may prevent the accurate analysis of fundamental statistical information such as mean daily runoff volumes, or the reliable interpretation of flow variability (Campozano *et al.*, 2014). Consequently, to avoid the effect of incomplete records on hydrological studies, and to make these studies more reliable, it is crucial to implement techniques that can perform estimations from incomplete records. According to Tencaliec *et al.*

(2015), this is termed the reconstruction, imputation or infilling of a dataset.

1.2 Objectives and domain of this study

This study proposes the application of modern data-driven modelling techniques, also known as machine learning, to address two hydrological problems discussed in Section 1.1: stream flow forecasting and gap infilling. Support vector regression and multilayer perceptron models will be considered, due to their popularity and applicability to various problems related to river basin management (Borji *et al.*, 2016).

The first objective of this study is to investigate single station short-term stream flow forecasting at a specific location in a river channel, by considering stream flow and precipitation time series records at that particular forecasting location. Three Australian river stations with sufficient time series records will be investigated. Support vector regression and multilayer perceptron models will be trained on the historical data of the stations to forecast stream flow with a lead time of up to seven days. The predictive capabilities of the machine learning models will be compared to that of a rainfall-runoff model provided by the Bureau of Meteorology (BOM), Australia's national weather and climate agency. They provide a forecasting service that supplies stream flow predictions at more than 100 locations across Australia. These forecasts are determined by a system which uses a rainfall-runoff model known as GR4J as its main component (Perrin *et al.*, 2003). This daily lumped, conceptual, four parameter, soil moisture accounting rainfall-runoff model determines the total amount of rainfall in a specific catchment, the fraction of rainfall that ends up as runoff, and the accumulation of that runoff in downstream rivers (Perrin *et al.*, 2003). Stream flow forecasts are given for a lead time of up to seven days (as shown in Figure 1.2), and are used for several water management purposes.

Secondly, we will investigate the ability of support vector regression and multilayer perceptron models to infill incomplete stream flow records. A particular case will be addressed where two different gauging stations are located along a river channel: one with an uninterrupted stream flow record, and the other one with gaps. The purpose of this part of the study is to infill the missing stream flow values of the one gauging station by considering the stream flow record of the other station, as well as data from any rain gauges within close proximity to the station considered for infilling.

The contribution of this study resides in the analysis of results. This includes the extent to which different environmental factors affect both machine learning and physically based model performances, which provides environmental researchers with insight into which climate variables may have a significant

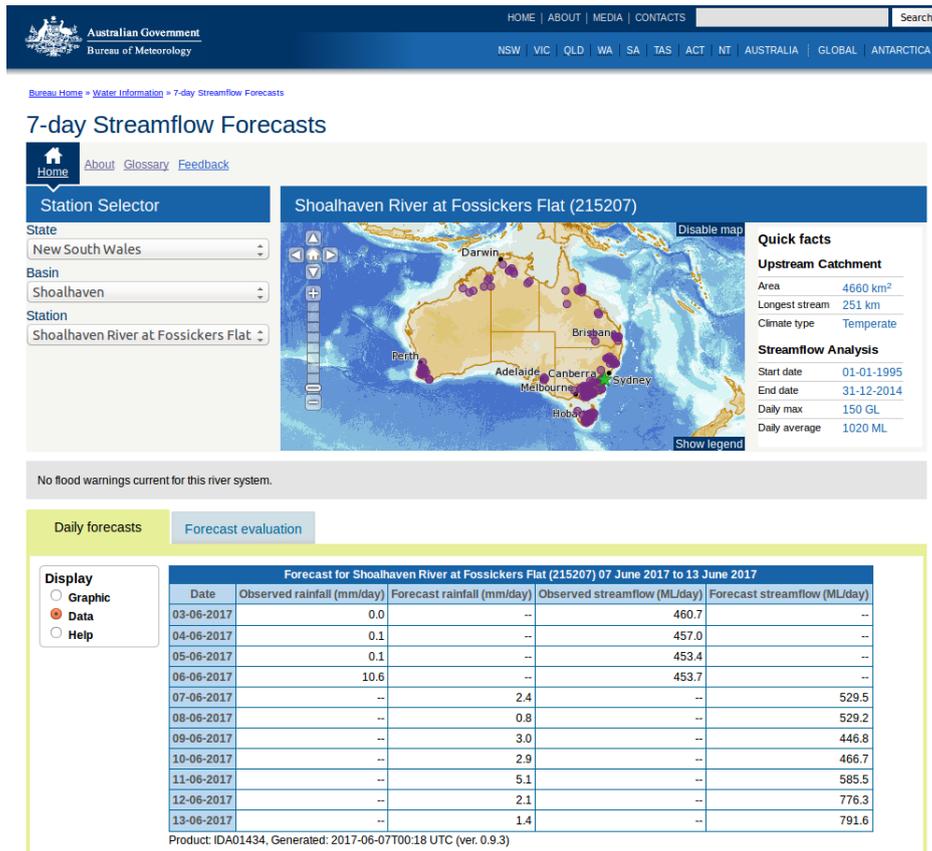


Figure 1.2: The forecasting service web portal provided by the Australian Bureau of Meteorology. Stream flow and rainfall forecasts with a lead time of up to seven days are provided to help river users in making decisions related to river and reservoir operations and water management (Bureau of Meteorology, 2017).

effect on stream flow. Furthermore, emphasis will be placed on good practices for machine learning system design in the field of hydrology.

1.3 Thesis layout

This thesis is organized into six chapters. Chapter 1 introduces current issues in the field of hydrology regarding stream flow modelling techniques and time series analyses. The rationale for the study, the objectives, scope and general research methodology are outlined. Furthermore, the publications from this study are listed.

Chapter 2 explains how hydrographs can be used to illustrate the effects of rainfall events on stream flow, and discusses the climatic and physiographic factors affecting their shape. Understanding the response of a given catchment's stream flow to rainfall input is helpful when choosing input features for the machine learning models, and may also give insight to the performance of

the models.

Chapter 3 provides a review of machine learning fundamentals, including training, validation, testing, data preparation and performance evaluation. A detailed description of the two modelling techniques considered for this study, namely support vector regression and multilayer perceptrons, is given and their advantages and drawbacks are outlined.

Chapters 4 and 5 describe techniques used in the development of support vector regression and multilayer perceptron models for short-term stream flow forecasting and gap infilling. Descriptions of the study areas and available datasets are given. Data analysis, feature selection, preprocessing and model performance evaluation techniques are discussed in detail. Furthermore, methods for choosing support vector regression hyperparameters and multilayer perceptron architectures are given. These chapters also present the results of the forecasting and gap infilling models, and their performances are discussed.

Chapter 6 presents concluding remarks based on our findings from Chapters 4 and 5. Recommendations for future research are also given.

1.4 Publications from this study

National conference paper

- Du Toit, M., Wilms, J.M., Smit, G.J.F. and Brink, W. (2016). The application of support vector regression (SVR) for stream flow prediction on the Amazon basin. *32nd Annual Conference of the South African Society for Atmospheric Science*, Cape Town, 31 October - 1 November 2016. ISBN 978-0-620-72974-1, pp. 25–28.

International conference paper

- Steyn, M., Wilms, J., Brink, W. and Smit, F. (2017). Short-term stream flow forecasting at Australian river sites using data-driven regression techniques. *4th International Work-conference on Time Series Analysis*, Granada, Spain, 18-20 September 2017. ISBN 978-84-17293-01-7, pp. 865–876.

Chapter 2

Stream flow and hydrographs

Stream flow or discharge is the volume of water that flows past a specific location in the river bed per unit time, and is usually measured at gauging stations situated along the river. Stream flow is a dynamic process that constantly changes due to various environmental factors. A hydrograph shows changes in stream flow at a specific location as a function of time and can be plotted in conjunction with a hyetograph (a graphical representation of rainfall intensity over time) to illustrate effects of preceding rainfall events, also referred to as storms, on stream flow.

Hydrologists assess the behaviour and performance of a hydrological model by estimating how well the observations made within the catchment are predicted. When considering stream flow modelling, a fundamental approach to evaluate model performance is through visual inspection of observed and forecasted hydrographs (Krause *et al.*, 2005). Hydrologists can assess whether the forecasted model over- or underpredicts actual stream flow, whether increasing and decreasing flow are accurately replicated, and whether the timing of the dynamic behaviour of the model is correct (Krause *et al.*, 2005). Since this approach will also be used to assess the results of our data-driven models, the main aspects of a hydrograph and the factors affecting its shape will be discussed.

2.1 Hydrograph shape

A hydrograph consists of two main components: base flow and overland flow. Base flow is the portion of stream flow supplied by groundwater. Overland flow is produced as a result of a rainstorm, and manifests in the form of surface runoff or through flow. Surface runoff is the water that flows directly over the land surface until it reaches the channel, whereas through flow is the lateral unsaturated flow of water in the soil zone which returns to the surface before entering the stream or becoming groundwater. The duration of

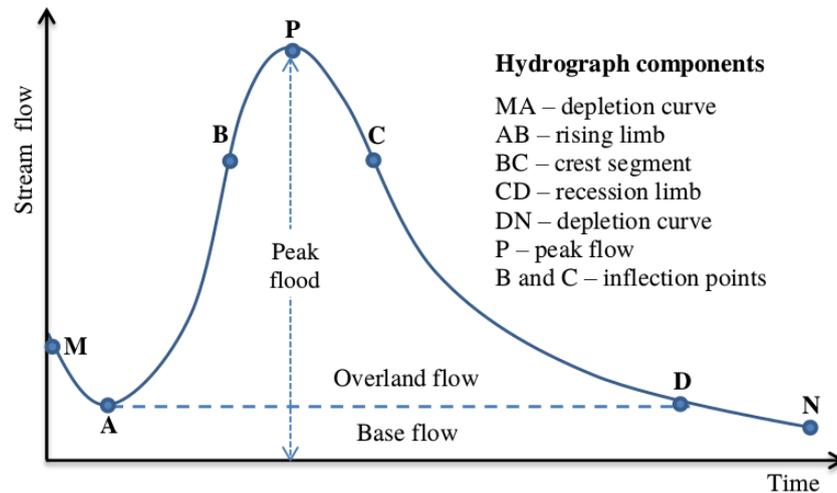


Figure 2.1: The main components of a typical hydrograph at a particular point in the river channel during a single storm, redrawn from Subramanya (2009).

overland flow is referred to as the hydrograph time base, and the total runoff obtained from overland flow is represented by the remaining area above the base flow on a hydrograph. The boundary between overland flow and base flow is dependent on the catchment structure and composition and may be challenging to determine.

The hydrograph shape represents the time distribution of runoff and follows a typical pattern when a single storm occurs over the catchment area. The main components of a hydrograph are the rising limb, the crest segment and the recession limb, as indicated in Figure 2.1 (Subramanya, 2009).

2.1.1 Rising limb

The rising limb of a hydrograph, represented by AB in Figure 2.1, describes a rise in stream flow as a result of channel and catchment surface storage slowly building up (Subramanya, 2009). During the initial stages of a storm, rainfall is first lost to processes such as interception and infiltration, causing a time delay before the rainfall excess reaches the stream and leads to a slow rise in stream flow (Wisler and Brater, 1959). The portion of rainfall contributing to stream flow is termed effective rainfall, whereas the remainder is evaporated, retained in the soil or detained on the land surface. A prolonged storm leads to an increase in effective rainfall, since infiltration losses decrease and more flow from distant parts of the catchment reaches the basin outlet (Subramanya, 2009). The slope of the hydrograph's rising limb therefore increases rapidly with time.

2.1.2 Crest segment

An important feature of a hydrograph's crest segment, represented by BC in Figure 2.1, is the peak flow, defined as the maximum flow at the basin outlet (Subramanya, 2009). For larger catchments, the peak flow may occur even after the storm has ended. The time difference between the effective rainfall's centre of mass to the peak flow of the hydrograph is referred to as the basin lag time, and is primarily determined by basin and storm characteristics (Subramanya, 2009). It is important in flood-flow studies to be able to determine the magnitude of a channel's peak flow as well as the time of its occurrence.

2.1.3 Recession limb

The recession limb, represented by CD in Figure 2.1, describes the depletion of storage, i.e. the removal of water from storage that accumulated in the basin during the beginning stages of the storm (Linsey *et al.*, 1949; Subramanya, 2009). Three main forms of water storage exist: surface storage (consisting of channel storage and surface detention), interflow storage, and groundwater or baseflow storage. The inflection point at the end of the crest segment, represented by C in Figure 2.1, corresponds to the basin's state of maximum storage (Subramanya, 2009). Storage depletion only occurs after the storm has ended. The shape of the recession limb is therefore dependent only on basin characteristics and not on storm characteristics (Linsey *et al.*, 1949). The relation between base flow and time is expressed by the lower part of a hydrograph's recession limb and is also known as the depletion curve (Wilson, 1974). The depletion curve is shown by DN in Figure 2.1, and indicates when the stream flow is entirely a result of groundwater seepage.

2.2 Ephemeral, intermittent and perennial rivers

A river may be classified as ephemeral, intermittent or perennial, based on the position of the catchment's water table (Roy *et al.*, 2009). Ephemeral streams consist of channels that are always above the water table. The existence of the stream is therefore completely dependent on effective rainfall. Streams that are seasonally dependent are defined as intermittent. The water table of an intermittent stream lies above the river bed during wet seasons and drops to a depth below the bed in dry seasons. During dry seasons, these rivers are dependent on effective rainfall for flow, whereas groundwater is contributed to the channels during the wet seasons. Perennial streams are river channels consisting of continuous flow throughout the year. The water tables of perennial streams are permanently above certain parts of the channel bed, constantly providing water to the stream. Characteristic hydrographs for the three types

of rivers are shown in Figure 2.2. Due to storm and basin irregularities as well as their complicated interactions, many of these hydrograph shapes may contain kinks and multiple peaks that differ from the simple single-peaked hydrograph in Figure 2.1.

2.3 Factors affecting a hydrograph

The shape of a hydrograph is dependent on many climatic and physiographic factors, also known as drainage basin controls. Table 2.1 lists the most important drainage basin controls, according to Subramanya (2009). Climatic factors mainly determine the rising limb, whereas physiographic factors affect the recession curve. A more detailed discussion on the main drainage basin controls and their effects on the hydrograph shape follows, assuming that the basin outlet is considered as the location where the stream flow is measured.

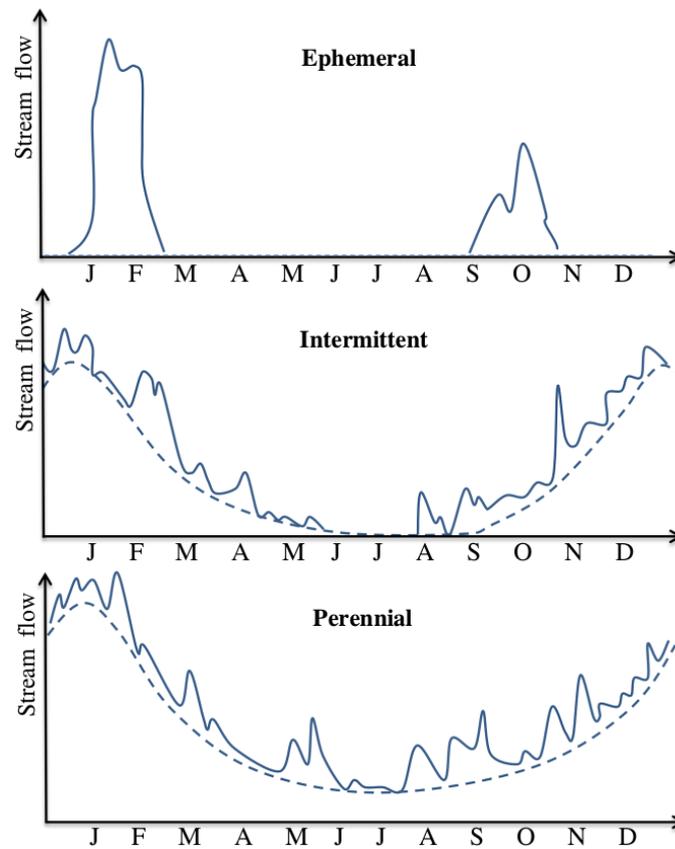


Figure 2.2: Typical hydrographs of the three types of rivers over a one year period. The dashed curves indicate the base flow of each type of river, whereas the solid curves show the hydrographs during storms.

Table 2.1: Climatic and physiographic factors affecting the hydrograph.

Climatic factors	Physiographic factors
1. Storm characteristics	1. Basin characteristics
(a) intensity	(a) size
(b) duration	(b) shape
(c) distribution	(c) slope
(d) direction	(d) drainage density
(e) type	(e) elevation
2. Evapotranspiration	2. Infiltration characteristics
	(a) land-use and vegetation
	(b) soil type
	(c) storage (lakes and swamps)
	3. Channel characteristics
	(a) cross-section
	(b) roughness
	(c) storage capacity

2.3.1 Climatic factors

The hydrograph shape and the amount of runoff that reaches the outlet are influenced predominantly by four climatic factors: the intensity, duration and distribution of a storm over the catchment, and the direction in which the storm moves.

Storm intensity

Storm intensity is defined as the amount of rainfall (in depth) per unit time and has an influence on the peak flow and the total volume of surface runoff for a given soil infiltration rate. A rainfall intensity that exceeds the soil's infiltration rate causes more overland flow and results in a steeper rising limb (Wisler and Brater, 1959).

Storm duration

Storm duration determines the peak flow and the duration of surface runoff, assuming a uniform storm intensity over the total catchment area (Wisler and Brater, 1959). An isochrone map consists of lines connecting points from which the runoff will take the same amount of time to reach the basin outlet, and may be useful in describing the effect of storm duration on the hydrograph of the catchment. Figure 2.3 illustrates a catchment where the point of measurement is at the outlet. When a storm occurs, the slope of the hydrograph's rising limb will start to increase. After a time Δt , the water from isochrone I would have reached the outlet and the whole area represented by A_I would be contributing to the rising limb. After a time period of $2\Delta t$, the water from isochrone II

would have reached the outlet and the whole area represented by A_I and A_{II} would contribute to the rising limb. If the rainfall continues until the entire catchment area contributes to the rising limb, the river is said to have reached its time of concentration (Linsey *et al.*, 1949). The hydrograph would reach a peak flow equal to $r_e A$, where r_e represents storm intensity and A the total area of the basin. The point of concentration may be reached in smaller catchments, and is therefore commonly used as the criterion for infrastructure development (such as bridges and culverts) and stormwater management (Saghafian and Julien, 1995).

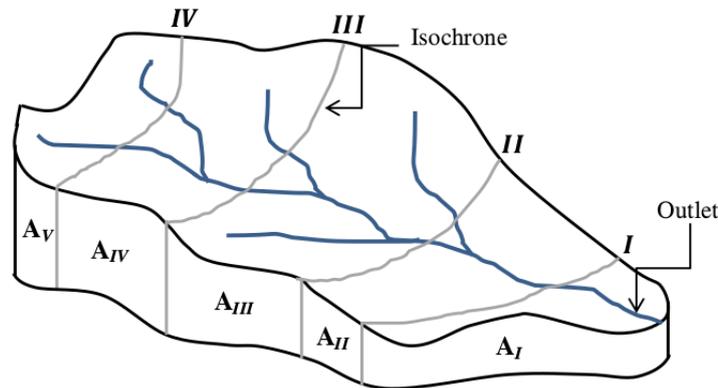


Figure 2.3: An isochrone map, consisting of lines (isochrones) that connect points from which the runoff will take the same amount of time to reach the basin outlet.

Storm distribution

The possible impact of storm distribution on the hydrograph shape can be explained by considering the isochrone map in Figure 2.3. If the storm is centred in an area near the basin outlet (such as A_I), the resulting hydrograph will show higher peak runoff compared to a storm centred in an area further away from the outlet (such as A_{IV}) (Linsey *et al.*, 1949; Wisler and Brater, 1959). According to Wisler and Brater (1959), rain that is uniformly distributed over a catchment produces the minimum peak runoff for a given total volume of rainfall and catchment characteristics.

Direction of storm movement

The direction in which a storm travels over a catchment with respect to the direction of river flow affects the resulting peak flow and the duration of surface runoff (Wisler and Brater, 1959). Elongated catchment areas are especially affected by the direction of a storm. If the point of flow measurement is considered to be at the outlet, a storm moving in an upstream direction would result in lower peak flow and a longer time base. Conversely, a storm moving

towards the downstream end leads to more rapid flow concentration at the outlet, resulting in higher peak flow and a shorter time base.

2.3.2 Topographic and geologic factors

The physical characteristics of a catchment are described by topographic and geologic factors and affect the shape of a hydrograph during a storm. These factors include the catchment shape, size, slope, drainage and land-use.

Catchment size

Smaller catchments show a different runoff behaviour compared to larger catchments, due to a difference in the relative importance of the existing runoff phases. In a smaller catchment, the overland flow phase has the greatest effect on the peak flow of the hydrograph, whereas the channel flow phase is more influential in a larger catchment (Subramanya, 2009). The time base of hydrographs for a smaller catchment will be shorter compared to that of a larger catchment, since water at the most remote point from the outlet has a shorter distance to travel.

Catchment shape

The shape of a catchment affects the time it takes water from the remote parts of the basin to reach the outlet point and therefore influences the resulting peak flow (Wisler and Brater, 1959). A catchment with a semi-circular or fan shape leads to a narrow and high-peaked hydrograph, whereas an elongated catchment shape gives a broad and narrow-peaked hydrograph (Subramanya, 2009). Figure 2.4 shows the effect of three different catchment shapes on a hydrograph, assuming identical rainfall and infiltration characteristics. Catchment A has a narrow end towards the upper basin area and a broader end towards the basin outlet, and will therefore result in a peak with a shorter drainage lag time. Catchment B has a broader end towards the upper basin area and a narrow end towards the basin outlet, and will therefore lead to a peak with a longer drainage lag time. Catchment C shows the hydrograph shape that results from a catchment with a composite shape.

Catchment and main stream slope

The slope of a catchment's main stream channel greatly affects the stream velocity and rate of storage depletion, and therefore alters the shape of a hydrograph's recession limb (Subramanya, 2009). A larger slope generates a greater velocity and causes more rapid storage depletion, resulting in a steeper recession limb and a smaller runoff time base. The catchment slope is essential in smaller catchments, since overland flow is more predominant than in larger

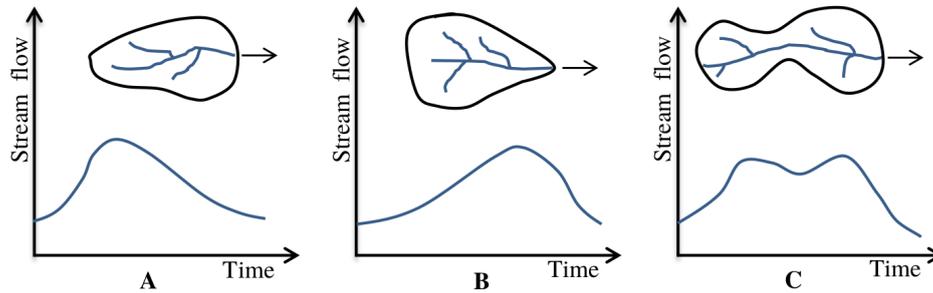


Figure 2.4: Effect of different catchment sizes on a hydrograph shape (Subramanya, 2009).

catchments (Subramanya, 2009). A catchment with a greater slope will therefore lead to larger peak flow values, compared to a catchment with a smaller slope.

Drainage

An important catchment characteristic is the arrangement of the natural stream channels in the area. Basins that are well drained allow a quicker disposal of runoff down the river, and causes a larger peak flow and a shorter drainage lag time compared to poorly drained basins (Wisler and Brater, 1959). Figure 2.5 illustrates the effect of a basin's drainage on the hydrograph shape, given that all other catchment and climatic characteristics remain identical.

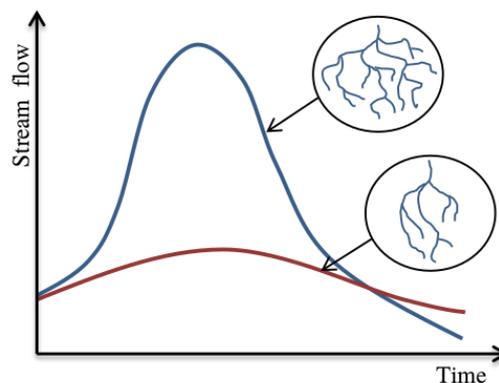


Figure 2.5: Effect of a catchment's drainage on the hydrograph shape (Subramanya, 2009).

Catchment land-use

The way in which the land within a catchment is utilised greatly influences the resulting peak flow (Wisler and Brater, 1959; Subramanya, 2009). Vegetation and forests intercept rainfall, increase the soil's storage capacity and delay overland flow. Therefore, less runoff reaches the river channel and over a longer period of time, resulting in a hydrograph with lower peak flow and a

longer drainage time lag and base. An urbanised area, consisting of impermeable surfaces such as roads and bridges, hinders the rainfall from infiltrating the surface. More rainfall reaches the channel as overland flow, resulting in increased peak flow and shorter drainage lag time.

Chapter 3

Data-driven modelling

Times series modelling is a dynamic research field that has evoked the attention of many research communities over the past few decades and has progressed significantly since 1970 (Adhikari and Agrawal, 2013; Box and Jenkins, 1970). Conventional linear models such as autoregressive moving average, autoregressive integrated moving average, linear regression and multiple linear regression models were developed and used for stream flow forecasting (Yaseen *et al.*, 2015). A drawback of these models, however, is their inability to adapt to nonlinear relationships in the data. Due to this limitation, more sophisticated data-driven modelling techniques were developed (Solomatine and Ostfeld, 2008).

Data-driven modelling is based only on data and is used to predict, but not necessarily explain, processes within a system. Developments in the area of machine learning have expanded the capabilities of data-driven modelling substantially (Solomatine *et al.*, 2008). Machine learning models analyse time series data to obtain functions that capture trends within the data. Numerous machine learning techniques have been applied to various hydrological processes, such as sediment transport, groundwater, water quality, precipitation, evapotranspiration, evaporation, floods, droughts and water levels (Yaseen *et al.*, 2015).

3.1 Fundamentals of machine learning

A central purpose of machine learning is to construct a model from historical data of the system under study, for the purpose of making predictions for that specific system from previously unseen data. The process of using available historical data to find a mapping between input and output data of a system is known as learning. As illustrated in Figure 3.1, the learning procedure aims to minimise the difference between actual observed output and predicted output (Solomatine and Ostfeld, 2008). When using machine learning to model a

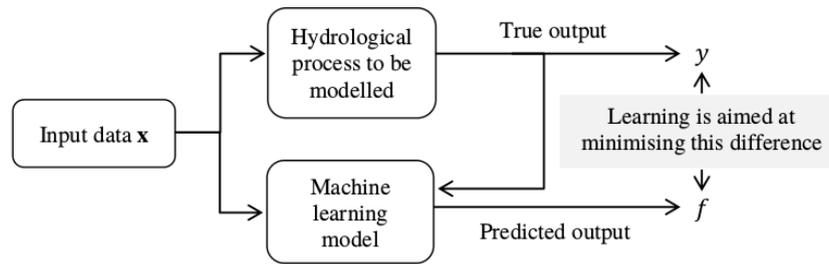


Figure 3.1: Supervised learning process, redrawn from Solomatine and Ostfeld (2008).

system, a sufficient amount of data characterising that specific system needs to be available.

Learning techniques can be distinguished based on the information available in the data. Specifically, if each available sample is a pair containing an input and output value (also known as a label), supervised learning can be considered. A supervised learning model aims to find the best target function f for the output y given its corresponding input \mathbf{x} , as illustrated in Figure 3.1. The given dataset is analysed, dependencies between the inputs and outputs are found, and a mapping $y = f(\mathbf{x})$ is constructed. After f is determined, it can be used for mapping new, previously unseen inputs. The process of learning a target function f is also known as training. Another type of learning can be used on datasets consisting of inputs and no corresponding outputs, and is called unsupervised learning. These methods analyse the dataset and cluster the data into different classes based on patterns or similarities found in the data.

A supervised learning model that predicts continuous variables is referred to as a regression model. If a model is used to categorise or predict discrete class labels, it is known as a classification model. Since hydrologically based problems are usually required to predict continuous variables such as stream flow or water levels, a regression model is considered.

3.1.1 Training, validation and testing

It is important to understand the methodology for using data when developing a machine learning model. A dataset is typically divided into three subsets, known as the training, validation and test sets. The data samples in the training set are used in learning a target function, as previously mentioned. During learning, the model complexity may increase to produce decreasing errors on the training data (Bray and Han, 2004; Solomatine and Ostfeld, 2008). However, when considering only the training set in the construction of a model, the problem of overfitting may occur. Overfitting is caused when the model learns the detail and noise in the training set to an extent that its performance on new unseen data (also referred to as the test set) is impacted

negatively. The model complexity is increased to fit the data samples in the given training set well, but large prediction errors are caused on unseen data. On the other hand, underfitting describes a model with low model complexity that can neither properly model the training data nor generalise to new unseen data.

We therefore aim to train a model that generalises well to unseen data, and do so by introducing the validation set. During training, the model is tested by fitting it to the validation set. At first, the error on both the training and validation sets will decrease as the model complexity increases. However, at a certain point the validation error may start to increase as an effect of overfitting. As illustrated in Figure 3.2, the training process should be terminated at this point.

A more sophisticated generalisation approach is known as K -fold cross validation, where the training set is split into K folds of equal size. Each fold is considered as a validation set once while the remaining $K - 1$ folds are used as the training set, as illustrated in Figure 3.3. The best trained model or a weighted combination of all trained models can then be used as the final model (Hastie *et al.*, 2009; Solomatine and Ostfeld, 2008). K -fold cross validation also maximises utilisation of the data, especially when only a small set of data is available (Maier and Dandy, 2000).

A further way of improving a model's ability to generalise to unseen data is to ensure that the training and validation sets are representative of the same population (Solomatine and Ostfeld, 2008; Maier and Dandy, 2000). For

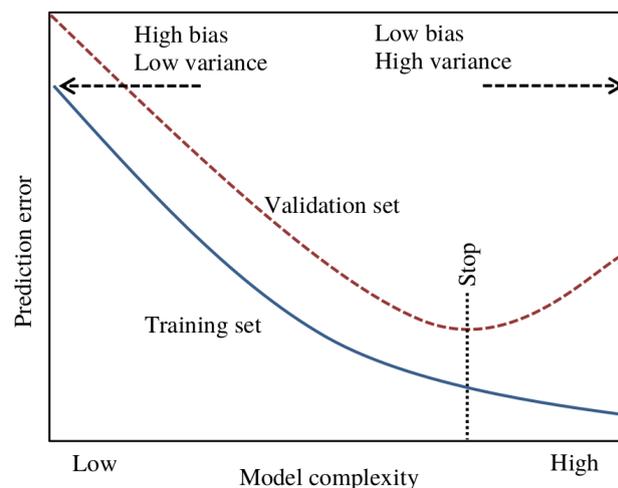


Figure 3.2: Generalisation of a machine learning model, redrawn from Bray and Han (2004). A model with low complexity may cause underfitting and result in large training and validation errors, whereas a model with high complexity may lead to overfitting by obtaining small training errors but large validation errors. A generalised model may be obtained by stopping the training process when a minimum validation error is obtained.

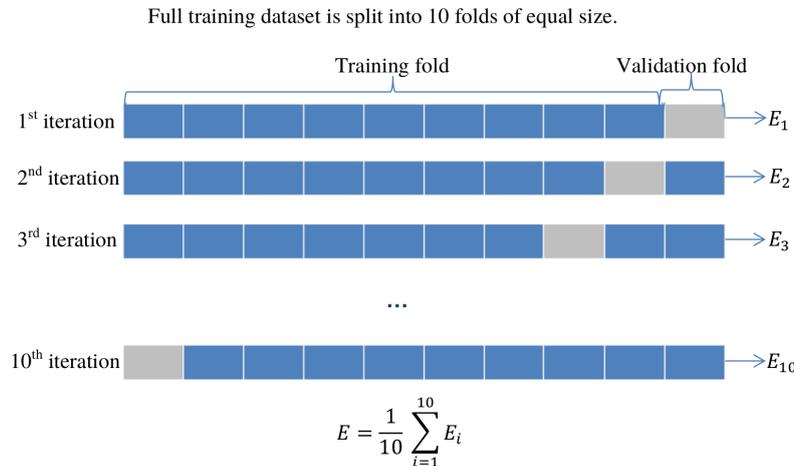


Figure 3.3: An illustration of 10-fold cross validation. The full training dataset is split into 10 folds of equal size. Each fold is considered as a validation set once, while the remaining 9 folds are combined to form a training set. Ultimately, the model with the lowest validation error on all 10 trials, or a weighted combination of all 10 models can be used for forecasting purposes. The error function E is computed as the sum of the squared difference between the true outputs and the network outputs.

instance, if a model is built to predict weather conditions of a specific region, the training set should contain data samples representing all four seasons of the year.

3.1.2 The bias-variance trade-off

The prediction error made by a machine learning model can be categorised as irreducible, bias, or variance. Irreducible errors are introduced in the model formulation of the problem and cannot be lessened by modifying the target function. Such errors may occur due to external factors that affect the way inputs are mapped to the outputs, but are not taken into consideration when constructing the model. For instance, consider modelling the water level of a dam, using rainfall measurements for that specific region. Due to the strong relationship between the input and output data, the resulting target function might be able to map the training data well. However, since other factors that influence the amount of water accumulating within the dam (such as temperature and evaporation) are not considered in the formulation of the model, an irreducible error might be present.

Bias indicates the difference between the expected prediction of a training model and the true value that it is trying to predict. Consider determining the target function for a set of data samples describing a specific process. By resampling the dataset, the model building process can be repeated many times to obtain an average model or target function. Bias refers to the difference between the predicted values of the average model and the true values. The

greater the difference, the higher the bias. A function with high bias may miss relevant relations between input and output data, and may therefore lead to errors when making predictions.

Variance refers to the amount of change in the predictions of the target function when considering different data samples. A model with high variance varies drastically from one dataset to another and may lead to unreliable predictions. High-complexity models may have high variance, since a small change in the dataset can cause a significant change in the shape of the target function.

An objective in constructing a supervised machine learning model is to obtain both low bias and low variance, since such a target function is likely to have good prediction performance. However, the difficulty in satisfying these requirements lies in the fact that a decrease in bias can lead to an increase in variance. Similarly, a decrease in variance can lead to an increase in bias. This is known as the bias-variance trade-off, and is also indicated in Figure 3.2.

3.1.3 Preparation of data

The preprocessing of data and the choice of variables that describe a modelled system can have a significant effect on model performance (Maier and Dandy, 2000; Solomatine and Ostfeld, 2008). One method of choosing variables is to rely upon the knowledge of a domain expert, who has an understanding of the hydrologic system that is being modelled (Bowden *et al.*, 2005). More formal methods have been developed to justify the choice of input features. For instance, linear cross-correlation is a popular analytical approach that determines the similarity between potential input features and the modelled process. If the similarity is strong, the considered feature may be a promising candidate for training the model. Imrie *et al.* (2000) built a stream flow forecasting model and considered cross-correlation analyses to determine appropriate lags of time series from upstream gauges as inputs. Various other examples of how cross-correlation have been used in hydrological studies are given by Bowden *et al.* (2005). However, a notable disadvantage concerning cross-correlation is its inability to capture nonlinear dependencies between inputs and outputs.

A commonly used heuristic approach for input feature selection is the stepwise technique. It is based on trial and error and considers different subsets of input. The two main stepwise approaches are known as forward and backward selection. Forward selection starts by finding the single best input feature for the final model. A set of selected input features are then considered, from which the feature that improves the model's prediction capabilities most is added to the final model. This process is repeated for all selected subsets of input features. Backward selection first considers all input features in a set. In each subsequent step, the input feature that reduces the performance the most is eliminated. Tokar and Johnson (1999) used the forward selection ap-

proach to find the input features for forecasting daily runoff in a watershed in the USA. A drawback of these heuristic approaches is that they can be computationally expensive. Furthermore, since they are based on trial and error, the globally optimal subset might not be found (Bowden *et al.*, 2005). Many other feature selection techniques have been developed and successfully implemented, including the stepwise partial mutual information algorithm (Bowden *et al.*, 2005) and the singular spectral analysis technique (Wang *et al.*, 2015).

Deep learning is a subfield of machine learning methods that is useful in circumventing the challenges of feature extraction. Deep learning models are able to learn how to extract an optimal feature vector for the given dataset using data compression algorithms known as autoencoding (Nezhad *et al.*, 2016). Autoencoders are especially useful when very large datasets are available for training.

For many machine learning models, data transformation is an important aspect of preprocessing. Two basic and widely used data transformation techniques are known as linear transformation and statistical standardisation (Shi, 2000). Linear transformation is often used in machine learning applications (Bowden *et al.*, 2003). The original data range is used to scale every dimension to a range of either $[-1, 1]$ or $[0, 1]$. This ensures that the influence of large feature values (like stream flow) does not dominate that of smaller feature values (like rainfall) during the training process. Statistical standardisation involves scaling the values of each input feature to have a zero mean and unit variance. For instance, consider an input vector \mathbf{Q} consisting of stream flow values. The standardised form of a particular stream flow feature Q is calculated as follows:

$$Q_{\text{standard}} = \frac{Q - \mu(\mathbf{Q})}{\sigma(\mathbf{Q})}, \quad (3.1)$$

where μ and σ represent the mean and standard deviation, respectively, of stream flow values in the training set.

3.1.4 Performance evaluation

Hydrologists assess the behaviour and performance of a hydrological model by estimating how well the observations made within the catchment are predicted. When considering stream flow modelling, a fundamental way of evaluating model behaviour performance is through visual inspection of observed and forecasted hydrographs (Krause *et al.*, 2005). As discussed in Chapter 2, hydrologists can assess whether the forecasted model over- or underpredicts observed stream flow, whether rising and falling limbs are accurately replicated, and whether the timing of the dynamic behaviour of the model is correct (Krause *et al.*, 2005).

The performance of a hydrological model can also be assessed by measuring the error between observed and forecasted variables. Three established methods

include Pearson's correlation coefficient, the root mean square error and the Nash-Sutcliffe efficiency.

3.1.4.1 Pearson's correlation coefficient

Pearson's correlation coefficient (r) gives the extent to which a model's predicted output and the true output are linearly correlated, and ranges between -1 and 1 . An r -value close to -1 or 1 shows a strong linear relationship between the two variables, whereas an r -value close to zero shows little to no relationship. If the predicted values of the model increase as the true values increase, a positive r -value is obtained. If the predicted values decrease as the true values increase (or vice versa), a negative r -value is obtained.

Mathematically, Pearson's correlation coefficient is expressed as

$$r = \frac{\sum_{i=1}^m (y_i - \bar{y})(f_i - \bar{f})}{\sqrt{\sum_{i=1}^m (y_i - \bar{y})^2} \sqrt{\sum_{i=1}^m (f_i - \bar{f})^2}}, \quad (3.2)$$

where y_i and f_i represent each of the m true and forecasted outputs in the test set, respectively. The average of all true outputs is represented by \bar{y} and the average of all forecasted outputs by \bar{f} .

3.1.4.2 Root mean squared error

The root mean squared error (RMSE) measures the difference between a model's predicted outcomes and the true outcomes from the system that is being modelled, and is expressed as

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - f_i)^2}. \quad (3.3)$$

The smaller the RMSE value, the better the performance of the model.

3.1.4.3 Nash-Sutcliffe efficiency

The Nash-Sutcliffe efficiency (NSE) is used to assess the predictive power of a model and is expressed as

$$\text{NSE} = 1 - \frac{\sum_{i=1}^m (y_i - f_i)^2}{\sum_{i=1}^m (y_i - \bar{y})^2}. \quad (3.4)$$

It is always less than or equal to 1. A model with an NSE of 1 corresponds to a perfect match of predicted outcomes to true outcomes. An NSE of 0 indicates that the model's predictive capability is the same as considering the mean true outcome value as a predictor. An NSE less than 0 occurs when the mean true outcome value would have been a more reliable predictor than the model itself (Krause *et al.*, 2005). According to Noori and Kalin (2016), a model can be considered "good" if the NSE is above 0.5, and "very good" if it is above 0.7.

3.2 Machine learning techniques in hydrology

According to Yaseen *et al.* (2015), the most extensively used machine learning models in the hydrological domain are neural networks, support vector regression, fuzzy logic, evolutionary computing and the wavelet transform. The research on and application of support vector regression and neural networks are the focus of this study, owing to their popularity and applicability to various problems related to river basin management (Borji *et al.*, 2016).

Neural networks have several advantages in hydrological forecasting, including their ability to model complex nonlinear processes such as the rainfall-runoff relationship (Wang *et al.*, 2015). The application of neural networks on hydrological forecasting studies have been widely used and published in recent years (Mehr *et al.*, 2015; Noori and Kalin, 2016). According to Bhagwat and Maity (2012), the application of support vector regression has also gained popularity in the field of hydrology. The advantage of a support vector regression model lies in the formulation of its convex objective function, ensuring that the global optimum may always be found. Furthermore, the resulting model provides a general solution that avoids overfitting, and nonlinear relationships can be modelled efficiently (Thissen *et al.*, 2003). Since the application on stream flow forecasting will be conducted using support vector regression and a neural network model known as a multilayer perceptron, a more comprehensive description on the formulation of these models follows.

3.3 Support vector regression

Support vector machines (SVMs) were introduced by Vapnik (1995) to solve machine learning problems and have drawn considerable interest in many research areas (Lee *et al.*, 2012). They were originally developed as a tool for solving classification problems, such as breast cancer diagnosis and bankruptcy prognosis (Lee *et al.*, 2012). An SVM constructs an optimal separating hyperplane that categorises data points into different classes. An optimal hyperplane is obtained when it has the best possible generalisation capabilities for unseen data samples and is constructed by solving an underlying optimisation problem over training data. SVMs have the ability to model complex data patterns through the use of a kernel trick that constructs nonlinear separating hyperplanes (Granata *et al.*, 2016; Lee *et al.*, 2012).

The SVM approach has also been extended to the task of regression and time series prediction, in the form of support vector regression (SVR). This technique generates a continuous-valued function that fits to the data samples in such a way that it shares similar advantages with SVMs. Many have addressed hydrological prediction problems using SVR (Dibike *et al.*, 2001; Granata *et al.*, 2016; Raghavendra and Deka, 2014; Thissen *et al.*, 2003).

3.3.1 Model formulation

Consider a training set of n real-valued data pairs $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$, where \mathbf{x}_i is an input vector of values in some space X , with corresponding output value y_i . The SVR model is used to fit a generalised continuous-valued target function $y = f(\mathbf{x})$ to the training set, such that a deviation of at most ϵ is obtained between each true output and its corresponding predicted value, and that $f(\mathbf{x})$ is as flat as possible (Granata *et al.*, 2016). Assuming f to be linear, we may write

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + c, \quad (3.5)$$

where $\mathbf{w} \in X$, $c \in \mathbb{R}$ and $\langle \cdot, \cdot \rangle$ denotes an inner product in space X . In order to get f as flat as possible, the orientation parameter (or weight) \mathbf{w} should be minimised. A quadratic convex optimisation problem can be constructed by minimising

$$\frac{1}{2} \|\mathbf{w}\|^2, \quad (3.6)$$

subject to constraints

$$-\epsilon \leq y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - c \leq \epsilon. \quad (3.7)$$

The objective function in (3.6) avoids overfitting of the target function by penalising larger weights. In (3.7) it is assumed that f can predict all pairs (\mathbf{x}_i, y_i) in the training set within an ϵ margin of error. However, some of the data pairs might exceed this margin and cause the optimisation problem to be infeasible. We introduce slack variables, denoted by ξ and ξ^* , which refer to the vertical distance to the data pairs above and below the ϵ margins, respectively. By penalising the slack variables based on their distance from the margins, the convex optimisation problem becomes one of minimising

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*), \quad (3.8)$$

subject to constraints

$$-\epsilon - \xi_i^* \leq y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - c \leq \epsilon + \xi_i, \quad \xi_i, \xi_i^* \geq 0. \quad (3.9)$$

The expression in (3.8) is known as the primal objective function. The positive penalty parameter C determines the tolerated deviations larger than ϵ . Predicted values outside the ϵ margin of error are penalised by the magnitude of the difference between the predicted values and the ϵ margin. This is also defined as the ϵ -insensitive loss function f_{loss} , and can be expressed as

$$f_{\text{loss}} = \begin{cases} 0, & \text{if } |\xi_i| \leq \epsilon, \\ C|\xi_i - \epsilon|, & \text{otherwise,} \end{cases} \quad (3.10)$$

for each data sample i . A graphical illustration of the ϵ -insensitive loss function is presented in Figure 3.4. Since the gradient of the function is determined by C , deviations are more severely penalised when a larger C -value is chosen.

The minimisation of equation (3.8), subject to constraints (3.9), is a standard constrained optimisation problem and can be solved by applying Lagrangian theory (Burges, 1998). A Lagrangian is constructed by multiplying each lower bound inequality constraint by a non-negative Lagrange multiplier and subtracting it from the primal objective function. This results in the following primal Lagrangian formulation:

$$L_P = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) - \sum_{i=1}^n \alpha_i (\epsilon + \xi_i - y_i + \langle \mathbf{w}, \mathbf{x}_i \rangle + c) - \sum_{i=1}^n \alpha_i^* (\epsilon + \xi_i^* + y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - c) - \sum_{i=1}^n (\eta_i \xi_i + \eta_i^* \xi_i^*). \quad (3.11)$$

The multipliers in (3.11) are represented by α_i , α_i^* , η_i and η_i^* . Multipliers without asterisks correspond to the training points above f and those with asterisks correspond to points below f . The primal Lagrangian function is minimised with respect to the primal vectors and variables (\mathbf{w} , ξ , ξ^* and c). A dual Lagrangian function L_D can be maximised with respect to the non-negative Lagrange multipliers. The Duality Theorem states that if an optimal solution exists for a primal problem when considering a convex objective function with a linear set of constraints, then the same optimal solution also exists for the

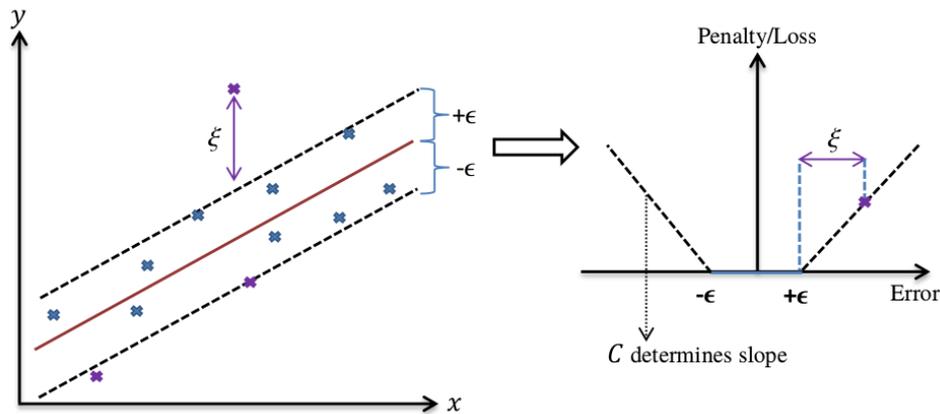


Figure 3.4: A linear support vector regression fit on data pairs with one-dimensional input vectors \mathbf{x} (horizontal axis) and corresponding output values y , redrawn from Thissen *et al.* (2003). Predicted values outside the ϵ margin are penalised in a linear fashion, as shown in the graph of the ϵ -insensitive loss function on the right. For this graph, the penalty parameter C determines the slope of the loss function, the horizontal axis represents the deviation of each predicted value from the true output, and the vertical axis represents the magnitude of the penalty.

dual problem (Bradley *et al.*, 1977). In other words, L_P has to be minimised with respect to the primal vectors and variables, while L_D has to be maximised with respect to all the Lagrange multipliers. An optimal solution can then be obtained.

A solution to the primal Lagrangian problem is obtained by determining the derivatives of L_P with respect to \mathbf{w} , ξ , ξ^* and c , and setting these equal to zero. The following conditions are obtained:

$$\frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^n (\alpha_i - \alpha_i^*) \mathbf{x}_i = \mathbf{0}, \quad (3.12)$$

$$\frac{\partial L_P}{\partial c} = \sum_{i=1}^n (\alpha_i^* - \alpha_i) = 0, \quad (3.13)$$

$$\frac{\partial L_P}{\partial \xi_i^*} = C - \alpha_i^* - \eta_i^* = 0, \quad (3.14)$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \eta_i = 0. \quad (3.15)$$

Substituting these conditions into the primal Lagrangian formulation in (3.11) yields the following dual optimisation problem:

$$L_D = -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \epsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*). \quad (3.16)$$

We maximise L_D by finding the optimal dual variables α_i and α_i^* , subject to constraints

$$\sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0, \quad (3.17)$$

$$\alpha_i, \alpha_i^* \in [0, C], \quad (3.18)$$

as derived from equations (3.13) to (3.15). By rewriting equation (3.12), the orientation parameter \mathbf{w} can be expressed as

$$\mathbf{w} = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \mathbf{x}_i, \quad (3.19)$$

which is a linear combination of the training data \mathbf{x}_i . Finally, by substituting equation (3.19) into equation (3.5), the target function f can be written as

$$f(\mathbf{x}) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \langle \mathbf{x}_i, \mathbf{x} \rangle + c. \quad (3.20)$$

Equation (3.20) is also known as the function's support vector expansion and describes the computation of a target function for linear regression purposes. Since the problem formulation is convex, the solution for f will always be globally optimal.

In order to determine the value of c in equation (3.20), the Karush-Kuhn-Tucker conditions are applied. Consider an optimisation problem of the following form:

$$\text{minimise } f(\mathbf{x}) \text{ subject to } h(\mathbf{x}) \leq 0. \quad (3.21)$$

Its Lagrangian is defined as

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda h(\mathbf{x}), \quad (3.22)$$

where \mathbf{x} represents a primal variable and λ represents a dual variable. Karush (1939) and Kuhn and Tucker (1951) state that for a local minimum \mathbf{x}^* there exists a unique dual variable λ^* such that

$$\nabla_{\mathbf{x}}(\mathbf{x}^*, \lambda^*) = \mathbf{0}, \quad (3.23)$$

$$\lambda^* \geq 0, \quad (3.24)$$

$$\lambda^* h(\mathbf{x}^*) = 0, \quad (3.25)$$

$$h(\mathbf{x}^*) \leq 0. \quad (3.26)$$

Equations (3.23) to (3.26) are known as the Karush-Kuhn-Tucker conditions. Equation (3.25) states that the product of the dual variables and the constraints should be set equal to zero. Therefore, referring to the primal Lagrangian function given by (3.11), it follows that

$$\alpha_i(\epsilon + \xi_i - y_i + \langle \mathbf{w}, \mathbf{x}_i \rangle + c) = 0, \quad (3.27)$$

$$\alpha_i^*(\epsilon + \xi_i^* + y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - c) = 0, \quad (3.28)$$

$$\eta_i \xi_i = (C - \alpha_i) \xi_i = 0, \quad (3.29)$$

$$\eta_i^* \xi_i^* = (C - \alpha_i^*) \xi_i^* = 0. \quad (3.30)$$

These conditions lead to some useful results. For instance, only the training points (\mathbf{x}_i, y_i) with $\alpha_i = C$ or $\alpha_i^* = C$ are located outside the ϵ margin of error. These points are known as the support vectors (as illustrated in Figure 3.5).

From equations (3.29) and (3.30), we see that $\xi_i = 0$ or $\xi_i^* = 0$ if $\alpha_i \in (0, C)$ or $\alpha_i^* \in (0, C)$, respectively. A solution for c can now be obtained by solving equations (3.27) and (3.28):

$$c = \begin{cases} y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - \epsilon, & \text{for } \alpha_i \in (0, C), \\ y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle + \epsilon, & \text{for } \alpha_i^* \in (0, C). \end{cases} \quad (3.31)$$

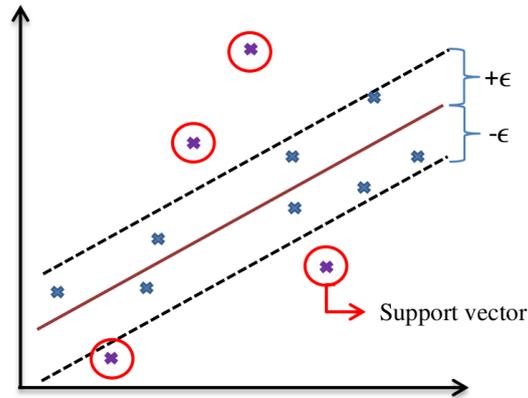


Figure 3.5: Support vectors are given by the encircled data points, located outside the ϵ margin. Redrawn from Raghavendra and Deka (2014).

3.3.2 Nonlinearity and kernels

The formulation of the support vector optimisation problem considered up to now assumes a linear relationship between the inputs and outputs in the training data. However, in many applications the relationship might be nonlinear. A kernel function k can be introduced to implicitly map the training points from the original input space X to a higher dimensional feature space $\Phi(X)$ such that a linear relationship between the variables exist in $\Phi(X)$. The support vector expansion of the target function for linear regression is then applicable in the feature space, as illustrated in Figure 3.6.

The linear support vector expansion given by equation (3.20) is expanded by mapping the input data \mathbf{x} from the original space X to some feature space

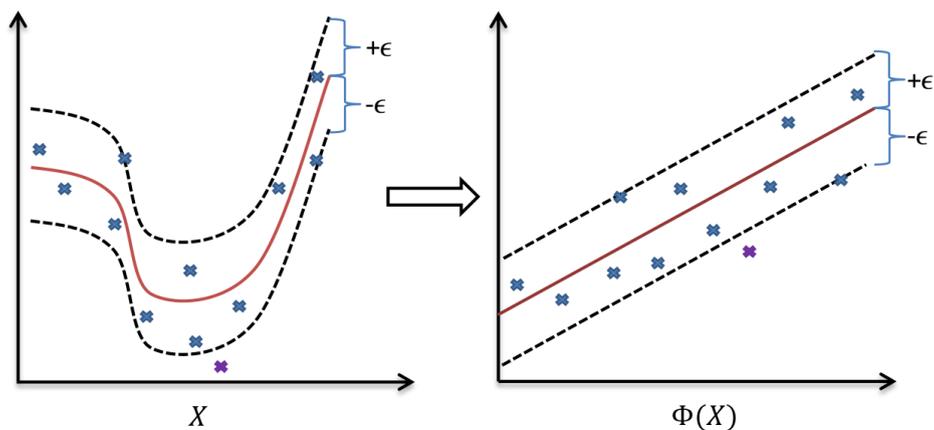


Figure 3.6: A nonlinear input-output relationship in the original space X on the left is mapped into the feature space $\Phi(X)$ on the right where the relationship becomes linear. The feature space is typically in a higher dimension. However, for illustration purposes, it is shown in the same dimension as the feature space. Redrawn from Thissen *et al.* (2003).

$\Phi(X)$. The solution of equation (3.20) changes to

$$f(\mathbf{x}) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) k(\mathbf{x}_i, \mathbf{x}) + c, \quad (3.32)$$

where

$$k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2) \rangle \quad \text{for } \mathbf{x}_1, \mathbf{x}_2 \text{ in } X. \quad (3.33)$$

As seen in equation (3.32), it is no longer required to find the flattest function in the input space X , but rather to find the flattest function in the feature space $\Phi(X)$. In SVR formulations, linear, polynomial, radial basis and sigmoid kernel functions are commonly used. These kernel functions are defined as follows:

$$\text{linear: } k(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{x}_1^T \mathbf{x}_2, \quad (3.34)$$

$$\text{polynomial: } k(\mathbf{x}_1, \mathbf{x}_2) = (\gamma \mathbf{x}_1^T \mathbf{x}_2 + r)^v, \quad (3.35)$$

$$\text{radial basis: } k(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma \|\mathbf{x}_1 - \mathbf{x}_2\|^2), \quad \gamma > 0, \quad (3.36)$$

$$\text{sigmoid: } k(\mathbf{x}_1, \mathbf{x}_2) = \tanh(\gamma \mathbf{x}_1^T \mathbf{x}_2 + r). \quad (3.37)$$

Variables γ , v and r are kernel-specific hyperparameters. The aim of optimising the SVR model's ability to generalise input data well is achieved by fine-tuning the model and its parameters (Bray and Han, 2004). Therefore, choosing an optimal model structure and corresponding hyperparameters, as well as values for ϵ and C , is important when training an SVR model to fit a given dataset (Granata *et al.*, 2016).

3.3.3 Advantages and drawbacks

As mentioned in Section 3.2, a significant advantage of a support vector regression model lies in the formulation of its convex objective function, which ensures that the global optimum will always be found. Furthermore, the penalty parameter C suppresses outliers within a dataset and therefore ensures a generalised model as well as robustness to noise (Bray and Han, 2004). According to Raghavendra and Deka (2014), another main advantage of SVR is the simultaneous minimisation of model complexity and prediction error by using the kernel trick.

The main drawback of SVR is the heuristic process of determining the optimal kernel function and corresponding hyperparameters, as well as the optimal values for ϵ and C (Raghavendra and Deka, 2014). This is usually determined by a grid search algorithm, which considers all possible parameter combinations, trains the SVR model on each combination and evaluates its performance using a metric such as K -fold cross validation. The optimal parameters are then chosen by determining the combination with the lowest cross-validation error. This can be a time-consuming and computationally expensive task, especially for larger grids.

3.4 Neural networks

Neural networks are parallel-distributed information systems consisting of a number of densely interconnected processing elements that work in unison to solve a specific problem (Yaseen *et al.*, 2015). A neural network can be designed for different types of applications, including pattern recognition and data classification. It has been extensively used for hydrological modelling purposes and time series forecasting applications, and has been found to be especially suitable when the underlying functions that describe complex phenomena are unknown (Maier and Dandy, 2000).

3.4.1 Model formulation

A neural network contains a set of interconnected nodes that receive, process and send information to one another over weighted connections. These nodes are grouped in different layers. As illustrated in Figure 3.7, input values enter the model through the first layer (the input layer). The data is then fed forward through successive hidden layers until it reaches the final layer (the output layer). The hidden layers enable the neural network to learn complex relationships between data (Solomatine *et al.*, 2008). A neural network can be single layered, bilayered or multilayered, depending on the number of hidden layers.

Neural networks are further classified as feed-forward or recurrent, based on the direction of information flow and processing between nodes. Feed-forward neural networks allow information to travel in only one direction: from the input layer to the output layer. Recurrent neural networks allow information to travel in both directions. Even though recurrent networks have shown to be very useful in time series applications, they are difficult to train and have a slower processing speed in comparison with feed-forward networks (Remesan

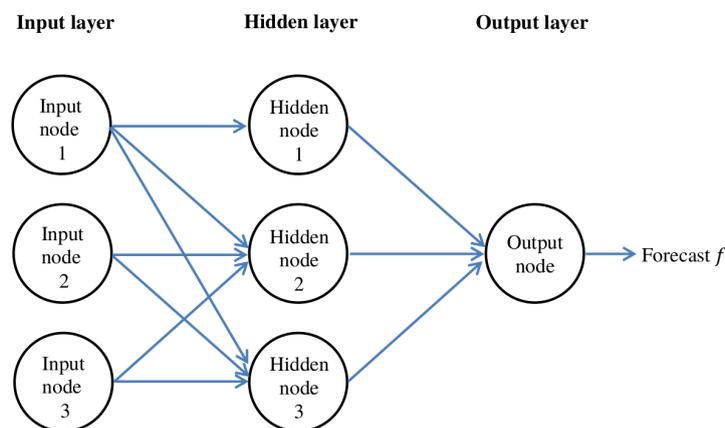


Figure 3.7: An example of a feed-forward neural network with one hidden layer.

and Mathew, 2014; Masters, 1993). According to Khotanzad *et al.* (1997), feed-forward networks have performed well compared to recurrent networks in many practical applications. Taver *et al.* (2015) also performed a study on the comparison of feed-forward and recurrent networks for non-stationary hydrological modelling and concluded that no model outperformed the other. Feed-forward networks will therefore be the focus of this study.

A widely studied and used feed-forward neural network model in hydrology is the multilayer perceptron (MLP). An MLP consists of an input layer, at least one hidden layer and an output layer. Weights determine how inputs are related to outputs and are assigned based on an input's relative importance to other inputs. For each node, an output is determined by calculating the sum of its weighted inputs, and applying a nonlinear transformation called an activation function. Furthermore, each layer contains an additional input with a numerical value of 1, for which its connected weight is known as a bias.

Consider the single hidden layered MLP given in Figure 3.8. Let i , j and k represent the position of each node in the input, hidden and output layers, respectively. Feed-forward computations are performed by first multiplying each input value x_i from the input layer with a set of connected weights w_{ij} connecting the input layer with the hidden layer. These weighted values are then summed with the bias b_j and transformed by the hidden layer activation function g_j , such that an output $g_j(b_j + \sum_i x_i w_{ij})$ is obtained for the j^{th} node in the hidden layer. Similarly, each output from the hidden layer is multiplied by the weights w_{jk} connecting the hidden layer with the output layer, summed

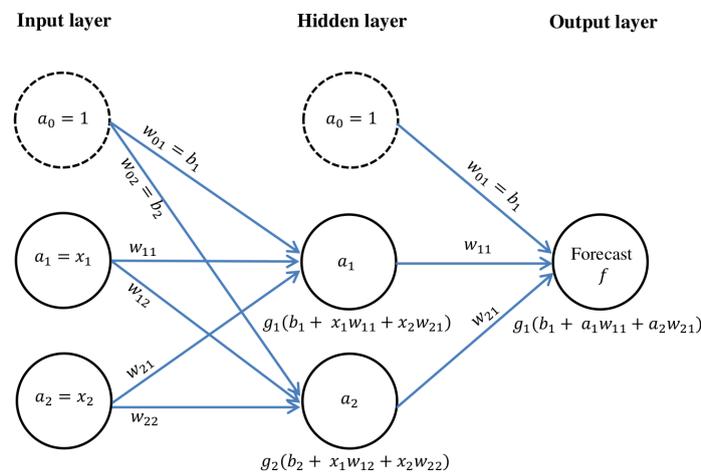


Figure 3.8: A single layered MLP. Feed-forward computations are performed by multiplying each input value x_i with a set of connected weights w_{ij} , connecting the input layer i with the hidden layer j . These weighted values are then summed with the bias b_j and transformed by the hidden layer activation function g_j , to obtain an output $g_j(b_j + \sum_i x_i w_{ij})$ for the j^{th} node in the hidden layer. A similar procedure is followed from the hidden layer to the output layer to obtain a final network forecast f .

with the bias b_k , and transformed by the output layer activation function g_k to obtain a final network forecast f .

As already mentioned, the activation function introduces nonlinearity in the input-output relationship of nodes by performing a mathematical operation on the sum of the particular node's input values. Choosing an appropriate activation function for the model is therefore an important task. According to Chang *et al.* (2007), Krishna (2014) and Maier and Dandy (2000), the sigmoidal-type and logistic sigmoidal-type (such as tanh) activation functions are frequently used in hydrological applications. These functions are as follows:

$$\text{sigmoidal-type: } g(z) = \frac{2}{1 + \exp(-2z)} - 1, \quad (3.38)$$

$$\text{tanh: } g(z) = \frac{1}{1 + \exp(-z)}, \quad (3.39)$$

where z represents the weighted sum of a specific node's inputs. This result is then used as input for the connected nodes in a succeeding layer. A linear activation function is often considered for the final hidden layer of regression models (Maier and Dandy, 2000).

Developing a neural network comprises of several processes. These processes consist of collecting, preprocessing and splitting data, establishing the model inputs, choosing the type and structure of the neural network, training the model to find an optimal set of connection weights for the training and validation sets, and testing the model on the remaining unused datasets. Data collection, appropriate preprocessing techniques, data splitting procedures (including cross-validation) and model input selection have already been discussed in Sections 3.1.1 to 3.1.4. A more comprehensive description of the remaining processes follows.

3.4.2 Network architecture

Designing a network architecture requires the determination of information flow direction, the number of hidden layers and the number of hidden nodes within each hidden layer. As already discussed in Section 3.4.1, feed-forward neural networks such as MLPs are long-established and popular for hydrological forecasting studies. According to Bhagwat and Maity (2012), De Vos and Rientjies (2005), Dibike *et al.* (2001) and Zealand *et al.* (1999), a one hidden layered feed-forward neural network provides suitable complexity to reproduce the nonlinear behaviour of hydrological systems and has been suitable for forecasting hydrological quantities in various studies.

It can be difficult to choose an appropriate number of hidden nodes within the hidden layer, as too few might result in a network that cannot capture the complex relationship between input and output, while too many may cause

overfitting. Panchal and Panchal (2014) give a review of different methods that have been used to select the number of hidden nodes. These include trial and error (Ghana Sheila and Deepak, 2013), rule of thumb (Karsoliya, 2012), simple (Karsoliya, 2012), two phase (Karsoliya, 2012) and sequential orthogonal approaches (Berry and Linoff, 1997). Belayneh and Adamowski (2013) proposed the combination of two different methods to use as bounds for the number of hidden nodes. Wanas *et al.* (1998) determined that the optimal performance of a neural network is obtained when $\log(n)$ hidden nodes are considered, where n represents the number of training samples. Mishra and Desai (2006) showed that optimal results are obtained for a neural network with $2N+1$ hidden nodes, where N represents the number of input nodes. Following Belayneh and Adamowski (2013), a trial and error approach can be implemented during training to find the optimal number of hidden nodes ranging between $\log(n)$ and $2N+1$.

3.4.3 Network training

The main purpose of training a neural network is to develop a model that replicates the input-output relationships of a specific system. This is achieved by finding the optimal set of connection weights and biases that minimise the error between the true output values and the output values that are determined by the network.

Optimisation is performed by considering either local or global methods. Local optimisation methods are the focus of this study and are classified as first-order or second-order, based on linear or quadratic models, respectively (Maier and Dandy, 2000). They produce computationally efficient methods of updating the weights using iterative techniques to minimise the error function. The weight update equation takes the general form

$$\mathbf{w}_{c+1} = \mathbf{w}_c + s_c \mathbf{d}_c, \quad (3.40)$$

where \mathbf{w}_c represents the vector of connection weights and biases, s_c gives the step size and vector \mathbf{d}_c specifies the direction of descent at iteration number c (Parisi *et al.*, 1996). The difference between the different local optimisation methods is determined by the choice of \mathbf{d}_c .

First-order local optimisation methods are based on the method of steepest descent, for which the descent direction \mathbf{d}_c is determined by the negative of the error function's gradient with respect to the vector of connection weights and biases. Equation (3.40) changes to

$$\mathbf{w}_{c+1} = \mathbf{w}_c - s_c \nabla_{\{\mathbf{w}_c\}} E, \quad (3.41)$$

where E represents the error function and is usually computed as the sum of the squared difference between the true outputs y and the forecasted (network) outputs f (Parisi *et al.*, 1996). Backpropagation is an extensively used

first-order method. A more detailed description of this method is given by Maier and Dandy (2000). A drawback of the backpropagation algorithm is its sensitivity to the initial conditions of the network, which causes it to easily get trapped in local optima (Maier *et al.*, 2010). According to Maier and Dandy (2000), the chances of finding a near-optimal local minimum improve when a number of networks are trained, each with a different set of initial weights. For large networks this approach could, however, become prohibitively expensive. Dropout is an alternative approach for addressing this problem. It provides a way of combining many different neural network architectures efficiently and prevents overfitting by randomly removing nodes along with their weighted connections from the neural network during training, thereby reducing strong dependency among specific nodes (Srivastava *et al.*, 2014).

Second-order local methods include the classical Newton method, the Levenberg-Marquardt approach, the quasi-Newton algorithm and the conjugate gradient method. The Newton method's weight update equation takes the form

$$\mathbf{w}_{c+1} = \mathbf{w}_c - \mathbf{H}^{-1} \nabla_{\{\mathbf{w}_c\}} E, \quad (3.42)$$

where \mathbf{H}^{-1} represents the inverse of the Hessian matrix (Parisi *et al.*, 1996). Drawbacks of this optimisation method include its expensive memory and computational requirements compared to first-order methods. Furthermore, the classical Newton method cannot ensure positive-definiteness of the Hessian, which is necessary for the optimisation algorithm to move downhill towards a minimum.

The Levenberg-Marquardt is a modified Newton method which ensures positive-definiteness of the Hessian matrix, but has the same expensive memory and computational requirements as the classical Newton approach (Parisi *et al.*, 1996). When the initial position is far away from a local minimum of the error surface, the algorithm behaves similar to a gradient descent method. However, when in close proximity of a local minimum, it has a quadratic convergence rate.

The quasi-Newton method sustains quadratic convergence while overcoming both problems associated with the Newton approach, by ensuring positive-definiteness of the Hessian matrix as well as reduced memory and computational requirements (Golden, 1996; Shanno, 1978). However, a limitation of the quasi-Newton method is its inability to escape local minima in the error surface.

Global optimisation methods, such as genetic algorithms, evolutionary programming and differential algorithms have an increased ability to overcome local minima in the error surfaces. However, their convergence speed and computational efficiency are worse compared to that of second-order local optimisation methods (Maier and Dandy, 2000).

3.4.4 Advantages and drawbacks

A main advantage of neural networks in hydrological applications is their ability to learn complex relationships between data, without any knowledge of the physical phenomena. According to Oyebode (2014), neural networks display structure compactness and flexibility and can be easily integrated into other data-driven modelling techniques. Furthermore, compared to some other modelling techniques, the computational requirements for feed-forward neural networks are relatively low.

As mentioned in Section 3.4.3, a drawback of neural networks is their sensitivity to the initial conditions of the network, causing them to easily get trapped in local optima (Maier *et al.*, 2010). Neural networks are also more susceptible to overfitting a dataset, compared to SVR models (Oyebode, 2014). Introducing a validation set or performing cross-validation during training is therefore essential to ensure that a generalised model is obtained. Furthermore, the optimal network design may differ for every modelling situation and is dependent on the specific dataset considered, which may make it challenging to decide on a suitable model structure (Oyebode, 2014).

Chapter 4

Single station short-term stream flow forecasting

Short-term stream flow forecasting refers to hourly or daily predictions and is vital for the implementation of a trustworthy water resources system (Raghavendra and Deka, 2014). In the past, three main types of hydrological models have been constructed for the purpose of short-term stream flow forecasting, and can be distinguished based on available information: conceptual, physically based and data-driven.

A conceptual model describes the scientific understanding of a system's current state. It does not describe the particular system using mathematical concepts, but rather gives sufficient information on every component of the model so that the system can be understood theoretically. In hydrology, conceptual models usually consist of a system of interconnected virtual tanks that are replenished and drained according to their dependencies on the processes within the hydrological cycle (Kokkonen and Jakeman, 2001). Many conceptual models have been constructed and used for modelling hydrological events, such as the Stanford Watershed model and the Tank model (Kokkonen and Jakeman, 2001). However, according to Brown (2012), they can be computationally expensive and often impractical when used to model the rainfall-runoff relationship for stream flow prediction within a river basin.

A physically based or process model is based on governing partial differential equations that describe the physical laws of a specific system. One of the first process models in the hydrological environment was developed by Freeze (1992). This model illustrates hillslope processes by solving the Richards equation using finite difference techniques (Kokkonen and Jakeman, 2001). A number of physically based models have since been developed to model hydrological processes, including the popular physically based rainfall-runoff models for stream flow prediction. However, as discussed in Chapter 1, one of the main challenges of these models is to determine an appropriate model structure and complexity for accurate simulation of hydrological behaviour at

catchment scale.

This chapter investigates whether machine learning models, specifically support vector regression and multilayer perceptron models, have the ability to overcome certain challenges faced by physically based models, and whether they have the potential to be useful tools for short-term stream flow forecasting. In the following section, the main objectives and methodology for this part of the study will be defined.

4.1 Methodology

The objective of this part of the study is to implement a machine learning model that can provide stream flow predictions with a lead time of up to seven days. Figure 4.1 is a schematic diagram describing the procedures to construct SVR and MLP models for stream flow forecasting at a specific river site. A discussion on these procedures follows.

4.1.1 Study area and data

High quality time series of daily stream flow and precipitation data for the Australian river sites under study were obtained from the Australian Bureau of Meteorology's Hydrologic Reference stations¹ (HRS) and Climate Data Online² (CDO) services, respectively. The HRS network consists of over 200 river sites that comply with the HRS selection criteria (Sinclair Knight Merz, 2010), namely to have a long period of high quality observations in catchments that are located in different hydro-climatic regions across Australia and are mostly unaffected by urbanisation and land-use change. The CDO service provides access to precipitation records from the Australian Data Archive for Meteorology.

According to Solomatine *et al.* (2008), machine learning techniques can be useful in modelling a hydrological system (or process within the system) if a sufficient amount of data describing the particular system is available and if the system has not changed significantly during the time period covered by the model. Land-use change and urbanisation may cause profound changes to the natural catchment conditions by changing the terrain, altering the soil and vegetation properties, and constructing buildings, pavements and water infrastructure. A discussion on how land-use change and urbanisation affect the rainfall-runoff relationship of a catchment is given in Section 2.3.2. It may also adversely affect the results of a hydrological machine learning model (Brown, 2012). As already mentioned, gauging stations in the HRS network are each mostly unaffected by water resource development and land-use change,

¹<http://www.bom.gov.au/water/hrs/about.shtml>

²<http://www.bom.gov.au/climate/data/?ref=ftr>

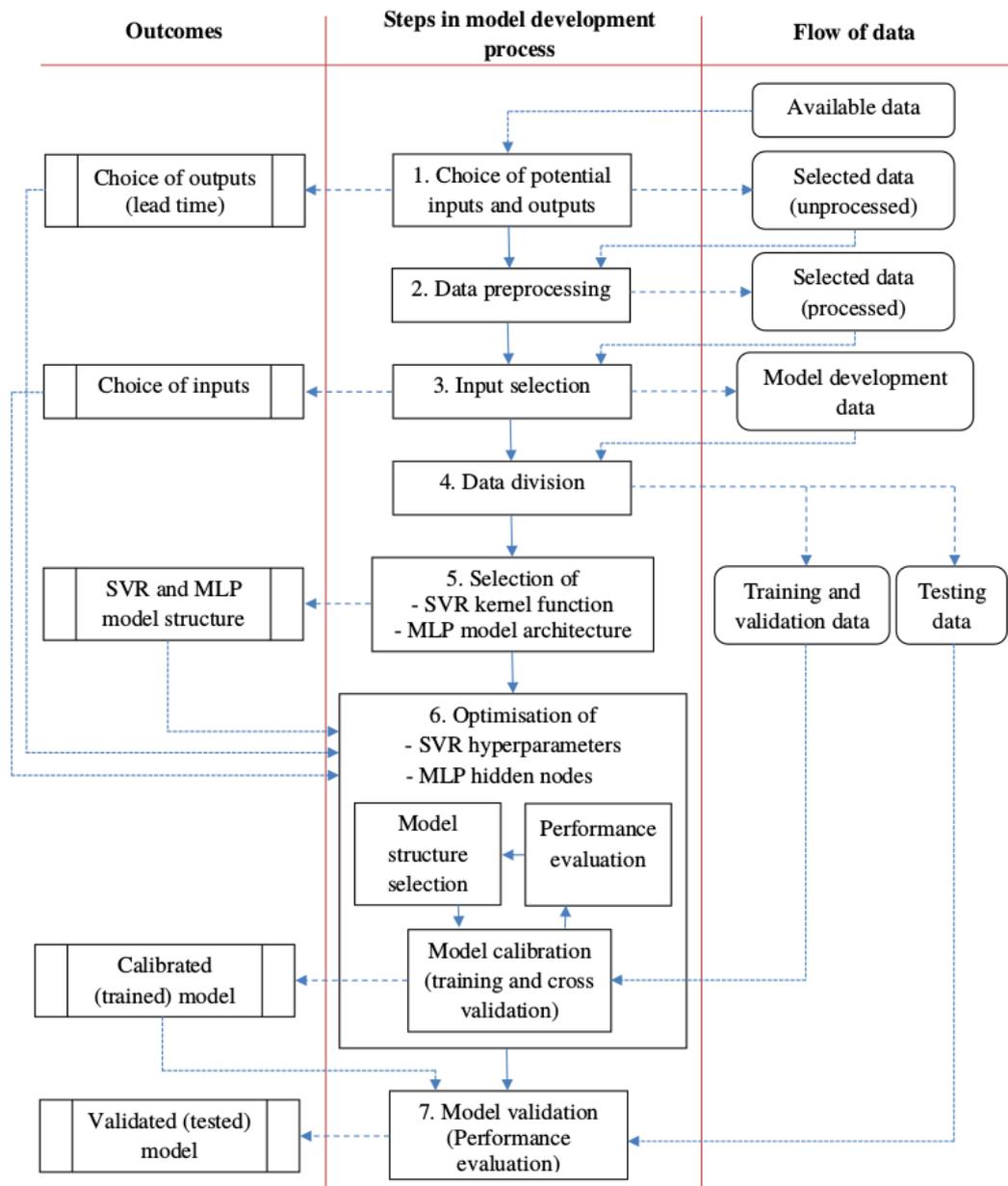


Figure 4.1: Steps in the SVR and MLP development process.

based on local knowledge of the catchment, stakeholder consultation and land-use analyses (Zhang *et al.*, 2016). These stations may therefore be reliable candidates for the implementation of machine learning techniques.

Three Australian river sites from the HRS network were considered for this study: the Shoalhaven River at Fossickers Flat in New South Wales, the Herbert River at Abergowie in Queensland, and the Adelaide River at Railway Bridge in the Northern Territory. Table 4.1 is a summary of the selected river sites, including the state, basin, location, climate, upper catchment area and the

Table 4.1: Summary of the three chosen river sites.

State	New South Wales	Queensland	Northern Territory
Basin	Shoalhaven	Herbert	Adelaide
Location	150.18° E 34.82° S	145.92° E 18.49° S	131.11° E 13.24° S
Climate	Temperate	Subtropical	Tropical
Upper catchment	4660 km ²	7488 km ²	638 km ²
Rainfall station	068085 (5.3 km from relevant gauging station)	032091 (8.7 km from relevant gauging station)	014237 (3.3 km from relevant gauging station)

corresponding rainfall station. Furthermore, the forecast locations are plotted on a map of Australia in Figure 4.2.

For this part of the study, only uninterrupted time series data were used for training: data from 1 January 2000 to 31 December 2014 for training the machine learning models at the Shoalhaven and Herbert rivers, and data from 1 January 2008 to 31 December 2012 for the Adelaide river. For all three river sites, data from 5 February 2017 to 5 August 2017 were used as test data.

Figure 4.3 shows the average daily precipitation and stream flow over the entire training periods of the three river sites, and can also be referred to as climatology graphs. Precipitation is measured in millimetres (mm), and stream flow

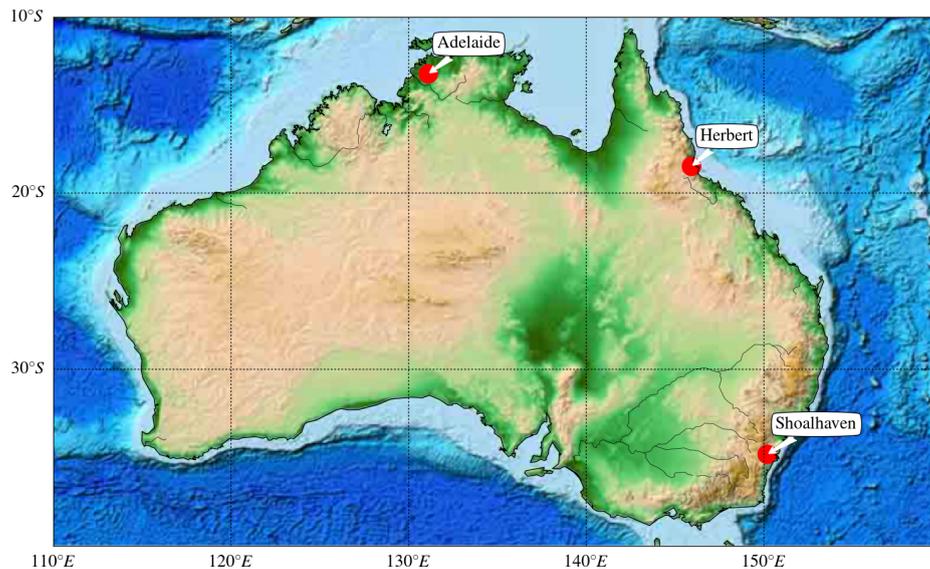
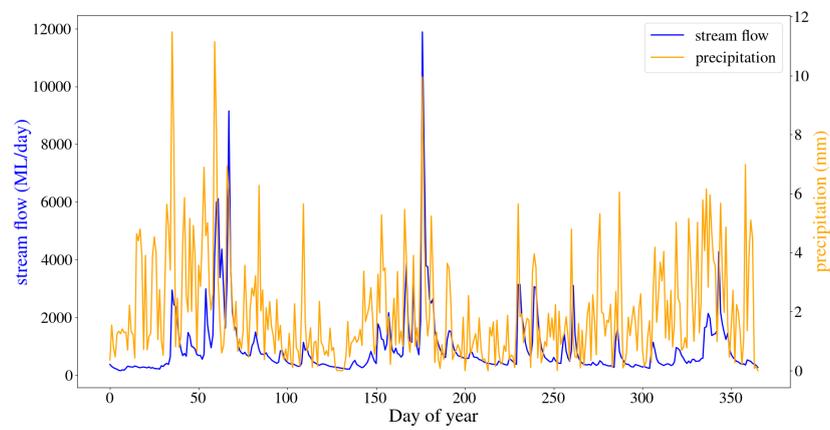
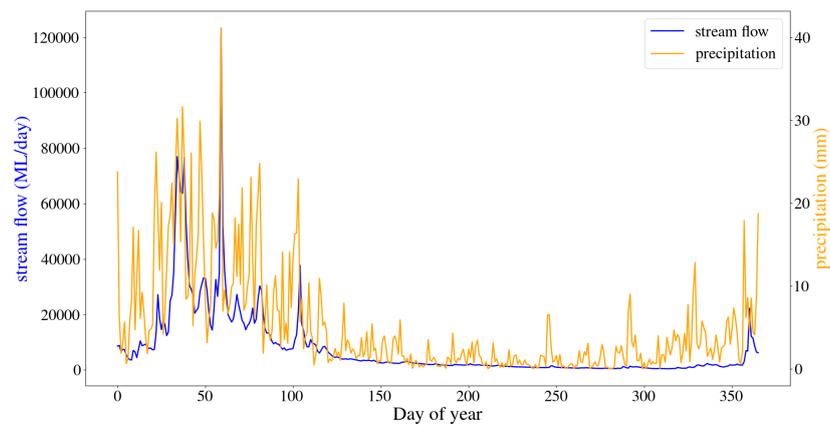


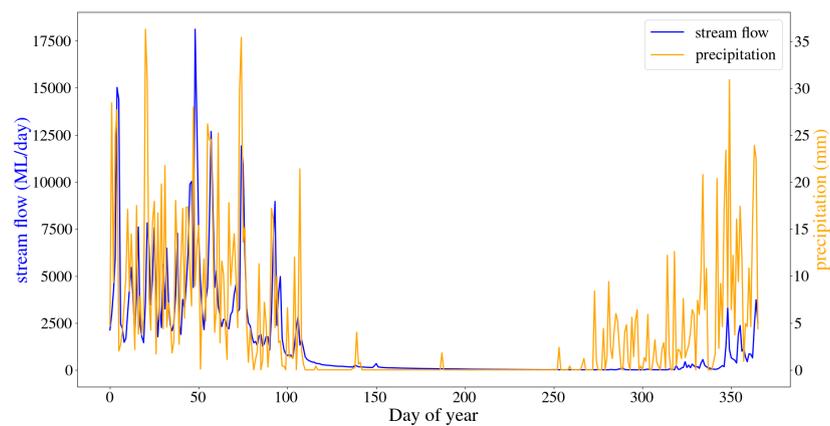
Figure 4.2: A map of Australia with the three forecast locations considered for this study: the Shoalhaven station in New South Wales, the Herbert station in Queensland, and the Adelaide station in Northern Territory.



(a) Shoalhaven



(b) Herbert



(c) Adelaide

Figure 4.3: Stream flow and precipitation climatology graphs of the Shoalhaven, Herbert and Adelaide river sites, hinting at a nonlinear relationship between precipitation and stream flow.

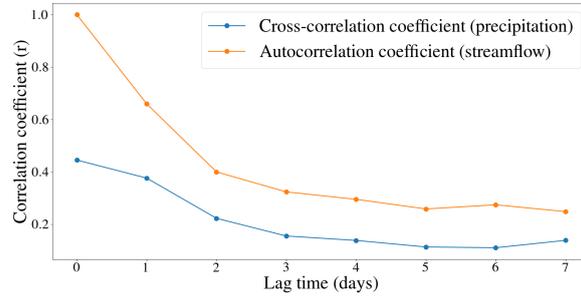
is measured in Megalitres per day (ML/day). According to Australia's Bureau of Meteorology, ML/day is a standard unit in irrigation and reservoir storage management applications. The streams at all three river locations are seasonally dependent, since the flow is higher during the wet seasons and low during the dry seasons. As discussed in Section 2.2, such streams may be classified as perennial, since they never run dry. According to the Australian Bureau of Meteorology, the tropical and subtropical zones (including the Adelaide and Herbert river sites) have distinct wet and dry seasons. The wet season, also known as the monsoon season, lasts from about November until March, whereas the dry season is usually between April and October. During the wet seasons, a high concentration of water in the air causes high humidity and high rainfall events and has caused severe floods in the past.

4.1.2 Selection of input variables

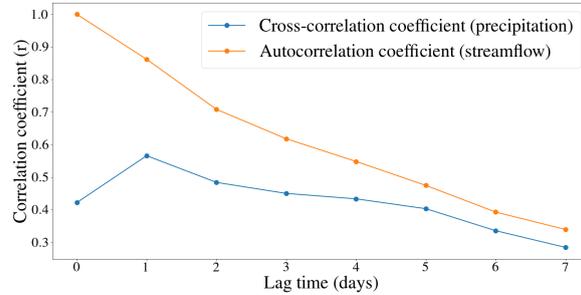
A moving time window was considered for the generation of input and output data pairs. For each measured stream flow value (which was considered as an output value), a corresponding input vector contained the precipitation and stream flow values of the preceding p -day and q -day time windows, respectively. P represents precipitation, Q represents stream flow, t refers to the current day and d refers to the forecasting lead time. An output value Q_{t+d} then had an input vector $\{P_t, P_{t-1}, \dots, P_{t-p+1}, Q_t, Q_{t-1}, \dots, Q_{t-q+1}\}$.

Selecting the appropriate number of lag times as input variables can be a difficult task. A visual inspection approach was followed by plotting a hydrograph in conjunction with a hyetograph for each river site, in order to determine the impact of preceding precipitation amounts on the stream flow. We refer back to Figure 4.3, where the average daily precipitation and stream flow values over the entire training periods of the three river sites are given. A visible relationship exists between the precipitation and stream flow data of the Herbert and Adelaide river sites, since the highest stream flow events occurred during the wet season, whereas the lowest stream flow events occurred during the dry season. This relationship is, however, less visible in the Shoalhaven data. Furthermore, it is difficult to establish the direct effect of precipitation lag on stream flow. Visual inspection by itself is not a good enough approach to determine the preceding precipitation to stream flow relationship at these specific river sites. Linear cross-correlation and autocorrelation were therefore also considered. As mentioned in Section 3.1.3, linear cross-correlation determines the similarity between potential input features and the modelled process. If the similarity is strong, the considered feature may be a promising candidate for training the model and should be included in the model building process.

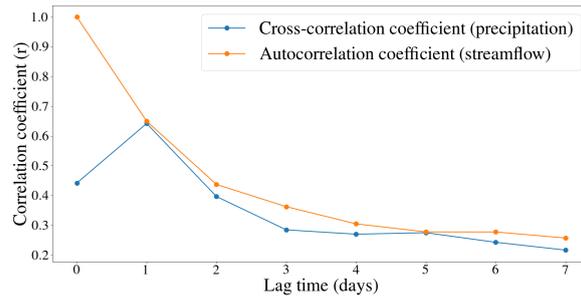
Consider a training set consisting of n data samples such that the precipitation set is given by $\{P_1, P_2, \dots, P_n\}$ and the stream flow set by $\{Q_1, Q_2, \dots, Q_n\}$.



(a) Shoalhaven



(b) Herbert



(c) Adelaide

Figure 4.4: Cross-correlation and autocorrelation for the Shoalhaven, Herbert and Adelaide stations, showing the strength of the linear dependence of precipitation with stream flow and stream flow with itself, respectively, considering a lag time up to 7 days.

Linear cross-correlation between $\{P_1, P_2, \dots, P_{n-l}\}$ and $\{Q_{1+l}, Q_{2+l}, \dots, Q_n\}$ can then be computed to determine the strength of the linear dependence between precipitation and stream flow, considering a lag time of l days. If the correlation is strong, then a model with output value Q_t might benefit from P_{t-l} as an input feature. Cross-correlation results for Shoalhaven, Herbert and Adelaide are shown in Figure 4.4. Lag times ranging from 0 to 7 days were considered.

The Shoalhaven station showed the highest rainfall to stream flow correlation when no lag time was considered, and showed a decrease in correlation as the lag time increased. The low correlation coefficients indicate a weak linear

correlation between the preceding rainfall and stream flow values. The Herbert and Adelaide stations showed the highest rainfall to stream flow correlations for a lag time of one day. Similar to the Shoalhaven station, an increase in the lag time lead to a decrease in correlation.

Autocorrelation between $\{Q_1, Q_2, \dots, Q_{n-l}\}$ and $\{Q_{1+l}, Q_{2+l}, \dots, Q_n\}$ was also computed to determine the strength of the linear dependence between stream flow and itself, considering a lag time of l days. If the correlation is strong, then a model with output value Q_t might benefit from Q_{t-l} as an input feature. Autocorrelation results for Shoalhaven, Herbert and Adelaide are shown in Figure 4.4.

All three figures show a decrease in correlation with an increase in lag time. Similar to cross-correlation, we may expect the machine learning models to assign a greater weight to stream flow values with shorter lag times. However, as mentioned in Section 3.1.3, a big disadvantage concerning cross-correlation is its inability to capture nonlinear dependencies between inputs and outputs. Therefore, even though the optimal linear cross-correlation and autocorrelation results were found for the shortest lag times, a stronger nonlinear correlation might be found for longer lag times. We therefore allowed the preceding p -day time windows to range from 1 to 3 and the q -day time windows from 1 to 5. Furthermore, to determine the necessity of including precipitation as an input feature, a model containing only preceding stream flow values was also trained, such that an output value Q_{t+d} had an input vector $\{Q_t, Q_{t-1}, \dots, Q_{t-q+1}\}$.

4.1.3 Preprocessing

Data preprocessing was implemented by linearly normalising the values in the training dataset to a range of $[0, 1]$. This ensured that the influence of large feature values (like stream flow) would not dominate that of smaller feature values (like rainfall) during the training process.

As discussed in Section 4.1.1, the available datasets were split into a training set and a test set. In order to obtain a model that generalises well to unseen data, 10-fold cross validation was introduced. The full training dataset was split into 10 folds of equal size. Each fold was considered as a validation set once, while the remaining 9 folds were combined to form a training set. Ultimately, the model with the lowest average validation error on all 10 trials was used for forecasting purposes, and tested on the test set (Solomatine *et al.*, 2008).

4.1.4 SVR hyperparameters

The SVR model with a radial basis kernel function, given by equation (3.36), was considered for this study. Three parameters had to be selected, namely

C , ϵ and γ . The C values ranged from 1 to 10^4 , ϵ values from 10^{-3} to 10^{-1} and γ values from 10^{-4} to 1 (all on a logarithmic scale). An exhaustive grid search was performed to find the combination of parameters with optimal performance during training and cross validation.

4.1.5 MLP architecture

As discussed in Section 3.4.2, a one hidden layered feed-forward neural network provides suitable complexity to reproduce the nonlinear behaviour of hydrological systems and was found to be suitable for forecasting hydrological variables in various studies. One hidden layer was therefore considered for the MLP models of this study.

Two different methods were used as bounds for the number of hidden nodes, as proposed by Belayneh and Adamowski (2013). As discussed in Section 3.4.2, a trial and error approach can be implemented during training to find the optimal number of hidden neurons ranging from $\log(n)$ to $2N+1$, where n represents the number of training samples and N the number of input nodes.

The sigmoidal-type and logistic sigmoidal-type activation functions, given by Equations (3.38) and (3.39), have been used frequently in hydrological applications. An exhaustive search was conducted to find the optimal function between these two.

Optimisation was performed by considering a second-order algorithm in the family of quasi-Newton methods, known as L-BFGS, which approximates the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm using a limited amount of computer memory (Skajaa, 2010). A more comprehensive description of the L-BFGS method is given by Skajaa (2010). A limitation of a local optimisation method, such as L-BFGS, is its inability to escape local minima in the error surface during training. Due to the random assignment of initial weights each time a network is trained, the starting points on the error surface may vary and may end up getting trapped in local minima, thereby resulting in different training errors. Therefore, to find an optimal set of input features, we considered 100 different random initializations and selected the preceding time windows for stream flow and precipitation that provided a minimum average training error.

4.1.6 Software

The stream flow forecasting models were implemented using scikit-learn, an open source machine learning library for the Python programming language. Specifically, the `sklearn.svm.SVR` and `sklearn.neural_network.MLPRegressor` functions were considered for SVR and MLP model development, respectively. Furthermore, the pandas Python package was considered for data manipulation and preprocessing purposes.

4.2 Results

Results of simulations for the optimal input features, hyperparameter combinations and model architecture for SVR and MLP models will be discussed in the following subsections. The predictive capabilities of our machine learning models will also be evaluated, based on the efficiency criteria given in Section 3.1.4.

4.2.1 Parameter selection

Seven different lead times were considered for stream flow forecasting, ranging from 1 day to 7 days in advance. As stated in Section 4.1.2, the preceding time windows for stream flow and precipitation that provided an optimal model were found separately during training for each of the different forecasting lead times. For SVR, an optimal combination of hyperparameters was also determined, whereas for MLP, an optimal number of hidden nodes and an activation function were selected. Results are listed in Table 4.2.

It can be observed that, when considering forecasts with different lead times, the preceding time windows for stream flow and precipitation and the combination of model parameters varied. It is also noticeable that only the MLP and SVR models for 3 and 7 day lead time forecasting of the Adelaide river site, respectively, and for 7 day lead time forecasting of the Shoalhaven river site obtained optimal results by not considering any preceding rainfall values. Apart from these particular cases, it appears that rainfall is an important input to the machine learning models for these three considered river sites. Furthermore, each MLP model achieved the lowest error during training and cross validation when the tanh activation function was applied.

4.2.2 Performance evaluation

The efficiency criteria used in this study were Pearson's correlation coefficient, the root mean squared error and the Nash-Sufcliffe efficiency. Based on these performance indices, the SVR and MLP models that performed optimally on the training and validation sets were applied to the (as yet unused) test sets of the three river sites. Results are shown in Table 4.3. For comparison, prediction accuracies made by the Bureau of Meteorology's stream flow forecasting model are also given. Furthermore, forecasts for only 1, 2 and 5 day lead times are shown in Figures 4.5 to 4.7.

Shoalhaven river site

The MLP model outperformed the SVR and BOM models for stream flow predictions at the Shoalhaven river site, as seen in Table 4.3a. The base flow as well as the rising and falling limbs of the hydrographs were well replicated

Table 4.2: Optimal input features and hyperparameters in the SVR and MLP models for the (a) Shoalhaven, (b) Herbert, and (c) Adelaide gauging stations (C , ϵ and γ are SVR parameters; h is the number of nodes in the MLP hidden layer; the model used precipitation data from days $t - p + 1$ to t and stream flow data from days $t - q + 1$ to t to predict stream flow on day $t + d$, with d the lead time). A dash sign signifies that no input from the specific process was considered as input.

(a) Shoalhaven

Lead time d	SVR					MLP		
	p	q	C	ϵ	γ	p	q	h
1 day	2	3	100	0.001	0.1	1	2	13
2 day	1	2	10	0.001	0.1	1	4	12
3 day	2	2	1	0.001	0.1	1	4	12
4 day	2	3	100	0.001	0.001	1	4	12
5 day	3	2	100	0.001	0.01	2	3	12
6 day	2	2	100	0.001	0.01	1	4	12
7 day	-	2	10	0.001	0.1	-	2	12

(b) Herbert

Lead time d	SVR					MLP		
	p	q	C	ϵ	γ	p	q	h
1 day	2	5	100	0.001	0.1	2	2	10
2 day	1	5	1000	0.001	0.1	1	3	10
3 day	3	5	100	0.001	0.1	2	5	13
4 day	1	4	10000	0.01	0.1	2	2	13
5 day	3	2	100	0.001	0.001	2	2	12
6 day	1	3	10000	0.01	0.1	1	3	12
7 day	1	3	10000	0.01	0.1	2	3	13

(c) Adelaide

Lead time d	SVR					MLP		
	p	q	C	ϵ	γ	p	q	h
1 day	2	5	1	0.001	1	2	2	11
2 day	1	5	1000	0.01	0.01	2	5	10
3 day	1	5	10000	0.01	0.01	-	4	10
4 day	2	5	10000	0.01	0.01	3	5	12
5 day	1	5	10000	0.01	0.01	2	3	11
6 day	1	2	10000	0.01	0.1	3	4	10
7 day	-	2	10000	0.01	0.1	2	2	10

Table 4.3: Performance evaluation for stream flow forecasting at the (a) Shoalhaven, (b) Herbert, and (c) Adelaide river station of our trained SVR and MLP models, as well as the physically based model used by the Australian Bureau of Meteorology (BOM).

(a) Shoalhaven

Lead time	Correlation (r)			RMSE			NSE		
	SVR	MLP	BOM	SVR	MLP	BOM	SVR	MLP	BOM
1 day	0.89	0.92	0.87	383	317	613	0.77	0.84	0.41
2 day	0.78	0.78	0.75	564	505	959	0.50	0.60	-0.44
3 day	0.73	0.75	0.63	662	553	1130	0.32	0.53	-0.98
4 day	0.69	0.68	0.58	735	612	1022	0.17	0.42	-0.61
5 day	0.57	0.52	0.35	764	701	2267	0.10	0.24	-6.91
6 day	0.48	0.48	0.23	802	719	3968	0.01	0.21	-23.15
7 day	0.41	0.44	0.26	824	734	2728	-0.04	0.18	-10.37

(b) Herbert

Lead time	Correlation (r)			RMSE			NSE		
	SVR	MLP	BOM	SVR	MLP	BOM	SVR	MLP	BOM
1 day	0.94	0.95	0.94	1357	1316	1557	0.87	0.88	0.83
2 day	0.83	0.84	0.91	2150	2162	1795	0.68	0.68	0.78
3 day	0.75	0.79	0.85	2651	2338	2397	0.51	0.62	0.60
4 day	0.70	0.71	0.80	2763	2699	2782	0.47	0.49	0.46
5 day	0.63	0.65	0.41	3630	2950	8918	0.08	0.39	-4.55
6 day	0.61	0.62	0.23	3032	3126	16178	0.36	0.32	-17.20
7 day	0.58	0.59	0.22	3163	3259	14675	0.31	0.27	-13.90

(c) Adelaide

Lead time	Correlation (r)			RMSE			NSE		
	SVR	MLP	BOM	SVR	MLP	BOM	SVR	MLP	BOM
1 day	0.86	0.89	0.88	680	694	642	0.73	0.72	0.76
2 day	0.72	0.76	0.67	1003	897	1014	0.40	0.52	0.39
3 day	0.68	0.67	0.54	949	1201	1152	0.44	0.11	0.18
4 day	0.62	0.63	0.39	1036	1111	1286	0.32	0.21	-0.05
5 day	0.58	0.65	0.29	1037	1263	1395	0.31	-0.02	-0.25
6 day	0.56	0.60	0.35	1052	1275	1267	0.29	-0.04	-0.03
7 day	0.46	0.55	0.43	1124	1318	1249	0.19	-0.11	0.00

by the MLP model for a lead time of up to 3 days. However, it is clear from Figure 4.5 that some of the peaks were estimated incorrectly. Furthermore, as seen during the month of May 2017, the MLP model predicted events of rapid increases in base flow, whereas in reality, the flow remained relatively unchanged. As the lead time increased, the MLP model overpredicted the base flow.

In comparison to MLP, the SVR model produced a slightly worse performance. As seen in Figure 4.5 the SVR model underpredicted the peak flows. However, it outperformed the other models with its accurate predictions of base flow. As the forecasting lead time increased, the accuracy of both the MLP and the SVR models decreased. Figure 4.5c shows that the SVR and MLP models failed to accurately forecast stream flow behaviour at the Shoalhaven river site for predictions with a lead time longer than 4 days.

The BOM model showed the worst performance in forecasting stream flow at the Shoalhaven river site in general. As seen in Figures 4.5b and 4.5c, the BOM model predicted steep rises and extreme peak flows during March 2017 and overpredicted the actual stream flow to a great extent. Furthermore, as seen in Figure 4.5, the base flow was underpredicted. Similar to the MLP and SVR models, accuracy of the BOM model decreased as the forecasting lead time increased. For 2 to 7 day lead time predictions, the NSE values of the BOM model were negative, indicating that the mean value of the observed outcomes would have been a more reliable predictor than the forecasting model.

Herbert river site

The BOM and MLP models showed the better performance on the test set of the Herbert river site when forecasting up to 4 days in advance. As seen in Table 4.3b, the MLP model outperformed the other models for a 1 day lead time forecast. In Figure 4.6a it can be seen that all models were able to replicate the stream flow behaviour (i.e. rising and falling limbs, peak flow and base flow) relatively well. However, the BOM model overpredicted the peak flow during March to a great extent. The BOM model furthermore outperformed the other models when forecasting with a lead time of 2 days. As seen in Figure 4.6b, the general stream flow behaviour was still predicted well, but the accuracy of the time and magnitude of peak flow decreased. Similar to the Shoalhaven forecasting models, an increase in forecasting lead time caused a decrease in model performance and an increase in lag times between observed peaks and forecasted peaks.

The SVR model did not perform as well as the other models in forecasting stream flow at the Herbert river site. The general stream flow behaviour was forecasted well for shorter forecasting lead times, but as the lead time increased, the forecasted peak flow values were generally too low and the peak flow times were delayed. Even though the SVR model obtained better NSE

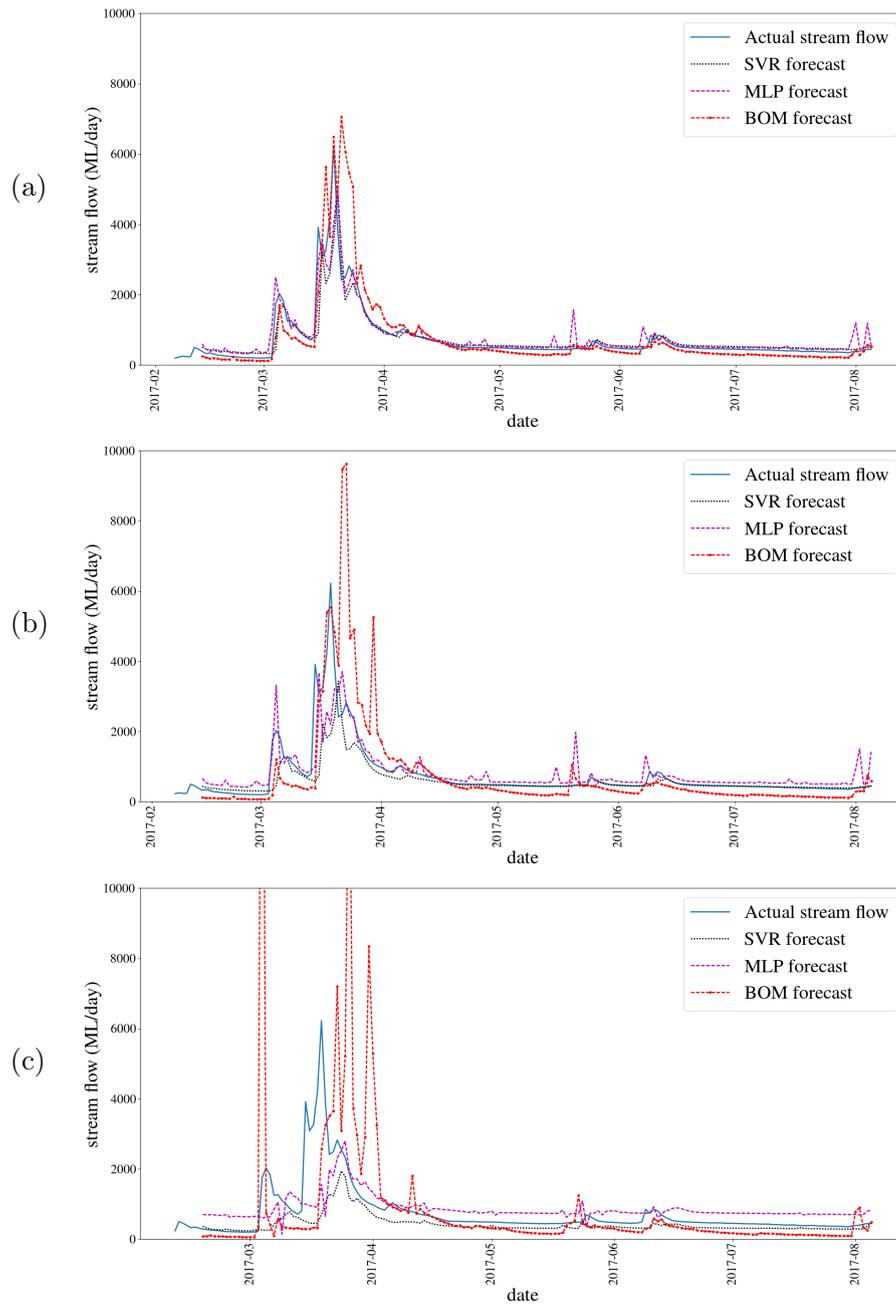


Figure 4.5: Daily stream flow predictions for (a) 1 day, (b) 2 day and (c) 5 day lead time forecasts, for the Shoalhaven station.

and RMSE values compared to the other models when forecasting 6 to 7 days in advance, it failed to determine accurate stream flow behaviour.

Adelaide river site

No single model outperformed the rest on the test set of the Adelaide river station. For instance, the MLP model obtained the strongest Pearson corre-

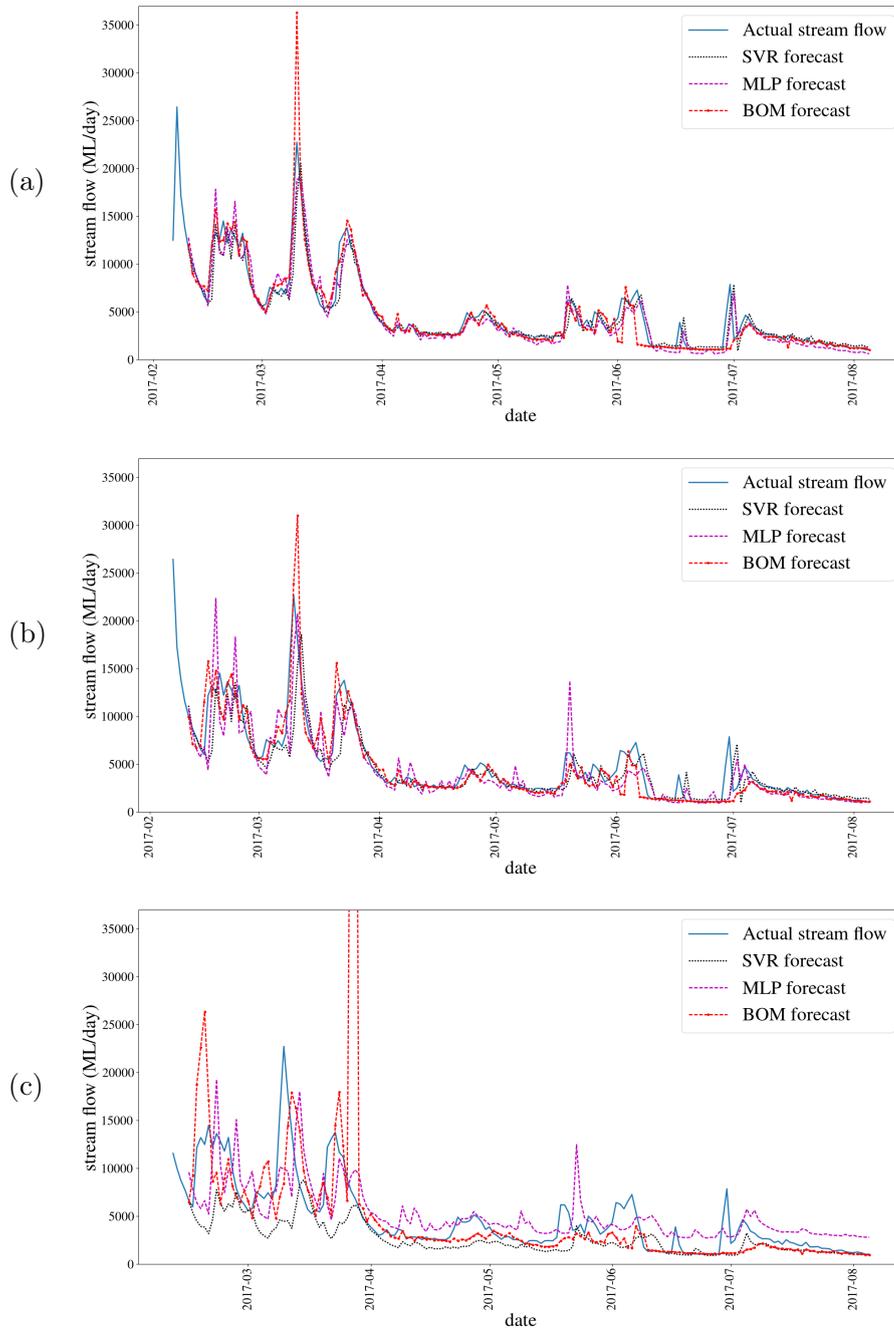


Figure 4.6: Daily stream flow predictions for (a) 1 day, (b) 2 day and (c) 5 day lead time forecasts, for the Herbert station.

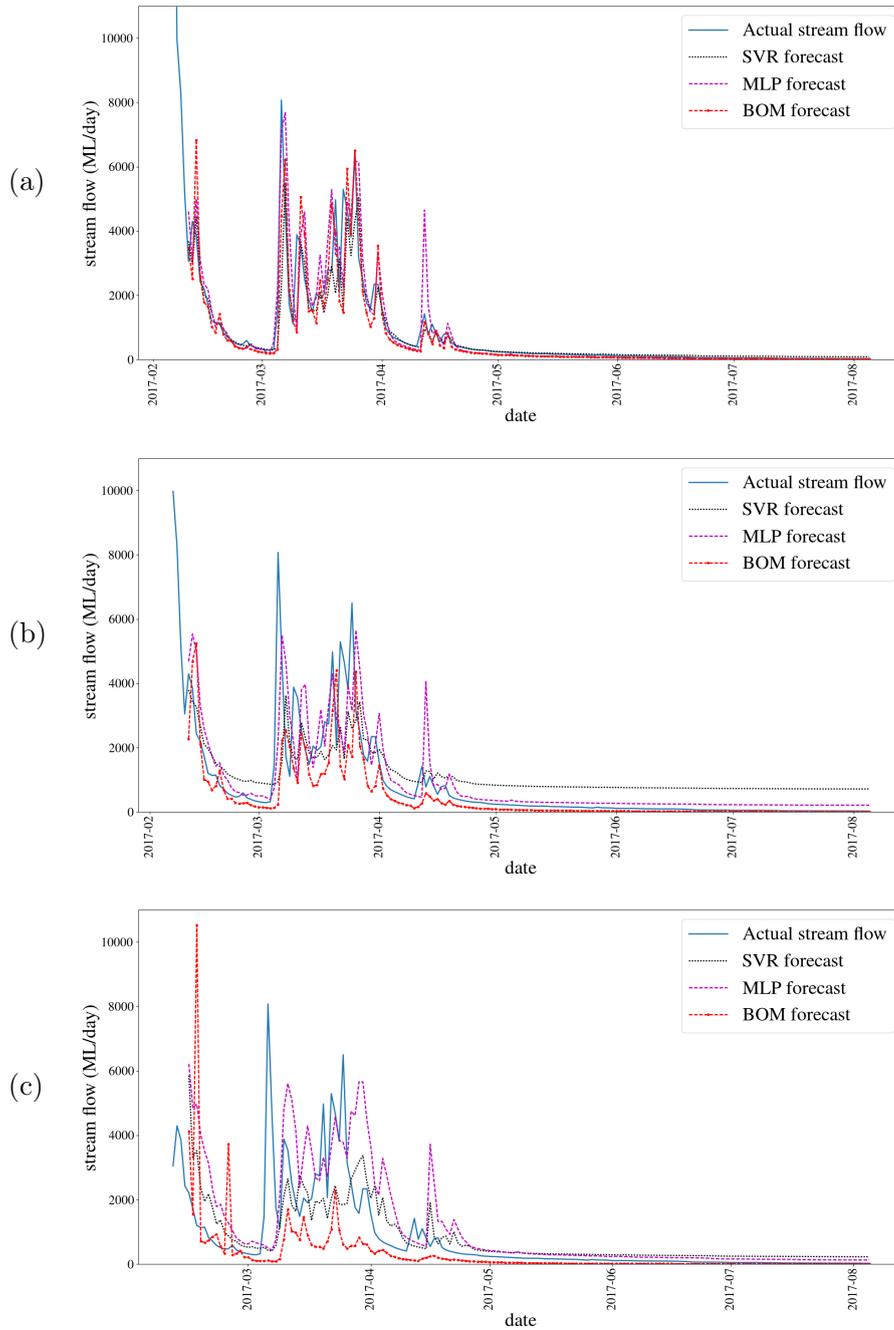


Figure 4.7: Daily stream flow predictions for (a) 1 day, (b) 2 day and (c) 5 day lead time forecasts, for the Adelaide station.

lation (0.89) to the observed stream flow when forecasting 1 day in advance, but failed to determine the flow magnitude as accurately as the BOM model. The MLP model did, however, show the better performance in forecasting stream flow with a lead time of 2 days, whereas the BOM model generally underpredicted the flow magnitude. The SVR model showed the better forecasting performance for predictions with a lead time greater than 2 days. As seen in Figure 4.7c, the observed stream flow behaviour was well replicated by the SVR model, whereas the MLP model generally overpredicted and the BOM model underpredicted the flow magnitude. Similar to both Herbert and Shoalhaven, the prediction capabilities of all three models worsened with an increase in forecasting lead time.

4.3 Discussion

Machine learning models as stream flow forecasting tools

Based on the results obtained for this study, MLP and SVR models have the potential to be useful tools for short-term stream flow forecasting. For a lead time of 1 day, each machine learning technique properly modelled the specific site's stream flow behaviour, and based on the evaluation criteria, the MLP model outperformed both the SVR and BOM models. Some of the peak flow events were, however, slightly misestimated, and a delay in some of the observed and forecasted peaks were visible. This may be attributed to the absence of important information about the catchment during training, leading to irreducible errors in the model. These errors may occur due to external factors that affect the way inputs are mapped to the outputs, but are not taken into consideration when constructing the model. There are several climatic and physiographic factors that affect the stream flow behaviour at a specific location in a catchment. As previously discussed, the intensity and duration of a storm influence the peak flow and the duration of surface runoff. The rainfall data used for this study only indicates the rain accumulation for the previous 24 hours, but does not give information about the intensity or duration of that rainfall. Furthermore, for this part of the study, stream flow and rainfall data at only one specific location were considered. Therefore, a rainfall event within close proximity to the considered gauging station, but not at the location of the considered rain gauge, may affect the stream flow behaviour but is not taken into account when developing the model. Considering data from more rainfall and upstream gauging stations within the catchment may improve these results.

Another possible cause for the delay in peak flow may be attributed to the absence of information between the most recent day on which input is considered and the day on which the forecast is made. For instance, rainfall events that occur after the input has been given to the forecasting model up to when the

forecast is made, may affect the stream flow behaviour but is not taken into account when developing the model. This also explains why forecasts with a longer lead time produced less accurate results. As mentioned in Section 3.1.4, good model performance is obtained when the NSE is above 0.5. For a lead time greater than 4 days, none of the forecasting models performed up to this standard. The BOM model did not even produce positive NSE values, indicating that this model's performance is worse than that of a simple model simulating a constant value equal to the mean value of the observed outcomes.

As mentioned in Section 2.1.2, it is important in flood-flow studies to be able to determine the magnitude of a river's peak flow as well as the time of its occurrence. Daily and weekly stream flow forecasts may be useful in making day-to-day decisions related to river and reservoir operations and management, but forecasts with a higher temporal resolution may give more accurate results regarding the magnitude and timing of peak flow for flood forecasting.

Bureau of Meteorology model

The physically based BOM model, provided by the Australian Bureau of Meteorology, generates forecasts on a daily basis, starting with real-time observations of rainfall and stream flow from a national network of rain and river gauges. These observations are integrated with the Bureau's rainfall forecasts to determine an estimated amount of runoff, as well as the flow of this runoff down the stream network. A forecast for each of the following seven days are then made. As seen in the results for the Shoalhaven, Herbert and Adelaide river sites, a high forecast skill was obtained for short lead times. This can be expected for 1 day lead forecasts, since only observed stream flow and rainfall values are taken into account. However, as the lead time increases, the rainfall forecast has a bigger influence on the stream flow forecast, and since the accuracy of the rainfall forecast also decreases as the lead time increases, the forecast skill of the model is likely to decrease rapidly. This can be seen especially in the results of the Shoalhaven station. For a 1 day lead forecast, the BOM model produced trustworthy results. However, for longer lead times it forecasted steep rises and extreme peak flows during March 2017 and over-predicted the actual stream flow to a great extent. This may be attributed to incorrect rainfall estimations. Table 4.4 shows the rainfall and streamflow forecasts made by the BOM model on 21 March 2017 for each of the following 7 days, as well as the actual measurements for those days. This table shows how the Bureau overpredicted rainfall, which in turn contributed to large errors in the stream flow forecasts.

Size of training set

An important aspect of machine learning system design is to obtain or define an acceptable training dataset. For this study, the largest available sets

Table 4.4: Forecasts made by the Bureau of Meteorology on 21 March 2017, compared with the true rainfall and stream flow observations. This table shows how the Bureau overpredicted rainfall, which in turn contributed to large errors in the stream flow forecasts.

Lead time	Forecasted rainfall (mm)	Observed rainfall (mm)	Forecasted stream flow (ML/day)	Observed stream flow (ML/day)
1 day	13.2	6.1	6065.5	2490.8
2 days	5.2	8.1	9634.8	2823.1
3 days	0.8	0.2	8779.6	2556.8
4 days	22.3	1.7	7227.8	2330.7
5 days	0.6	0	15480.5	1878.2
6 days	12.7	1.3	11268.8	1545.4
7 days	0.6	0.2	10527.9	1341.3

of coinciding rainfall and stream flow time series data were considered. The Shoalhaven and Herbert stations had 15 years of available data, whereas the Adelaide station only had 5 years. Even though it is uncertain how the Adelaide forecasting model would have performed on more training data, the existing model did not perform noticeably worse compared to the other models for which more training data were available.

Choosing a large enough training set is dependent on many factors, including the complexity of the system that is being modelled, as well as the learning algorithm. As discussed in Section 3.1.1, the data needs to be representative of the system that is being modelled, i.e. there needs to be a sufficient amount of data to reasonably capture the existing relationships, both between inputs and between inputs and outputs. If the relationship between stream flow and rainfall is nonlinear, a high-complexity learning algorithm may be required to model the system. As mentioned in Section 3.1.2, high-complexity models may exhibit higher variance, since a small change in the dataset may cause a significant change in the model. A higher complexity system may therefore require more training data to prevent overfitting.

A general principle in machine learning practice is to consider more training data, rather than less. However, as mentioned in Section 4.1.1, machine learning techniques can be useful in modelling a hydrological system not only if a sufficient amount of data describing the system is available, but also if the system has not changed significantly during the time period covered by the model. For instance, consider a river basin for which 20 years of stream flow and precipitation data are available, and assume that a large area of the basin is deforested for urbanisation purposes during the 5th year of the available data. The relationship between stream flow and precipitation will change significantly after this event, and therefore only the remaining 15 years' data can be used for training (given that no other land use or land cover change event

occurred during this period).

Information about the history of a catchment is not always available, and alternative ways of determining changes in the system may be required. One approach is to consider a double mass curve, which provides a way of determining changes in the relationship between rainfall and stream flow of a catchment. This is done by considering a graph of the cumulative rainfall versus the cumulative stream flow over the training period of the catchment. A perceptible change in the slope of the line may be linked to a particular event or series of events that changed the relationship between rainfall and stream flow. The time step of accumulation depends on what type of system change one might be interested in. Daily accumulation could represent responses to particular rainfall events, monthly accumulation could represent changes in seasonality of catchment responses, and annual accumulation could represent the effect of longer cycles on the catchment, such as the drought in Australia during the 2000s. This approach is, however, often used as an exploratory tool rather than a rigorous test.

A well-established statistical method for determining changes in a catchment is the paired catchment approach. It can be used to analyse the effects of catchment changes, also referred to as treatments, on stream flow. These treatments include land use, urbanisation and the construction of dams. The paired catchment approach is based on the assumption that functional relationships exist between the stream flow variability of two basins in close spatial proximity (Salavati *et al.*, 2016). Since the climate conditions for both catchments are assumed to be similar, the functional relationships between the two remain justifiable for as long as the catchments remain undisturbed. However, a change in relationship occurs when one catchment (referred to as the disturbed or treated catchment) experiences changes such as urbanisation, while the other catchment (referred to as the control catchment) remains undisturbed (Salavati *et al.*, 2016). A limitation of this method is the lack of available, undisturbed control catchments near a treated catchment. Usually, the control catchment either has also undergone changes or is too far from the disturbed catchment to assume climatological similarity (Salavati *et al.*, 2016).

A limitation of machine learning in hydrology is that substantial historical stream flow and precipitation records should be available for training. For this study, uninterrupted sets containing at least 5 years' data were available for training. However, many existing gauging stations have limited datasets, or a considerable amount of missing data. Incomplete datasets increase the complexity and uncertainty of hydrological modelling, and even very small gaps may prevent the accurate analysis of fundamental statistical information such as mean daily runoff volumes, or the reliable interpretation of flow variability (Campozano *et al.*, 2014). It is therefore crucial to implement techniques for the estimation of incomplete records. In the following chapter, we will aim to

address this problem using SVR and MLP as infilling techniques. Specifically, catchments containing more than one gauging station will be considered, to determine whether information from one station can be used to infill gaps in the stream flow record of another station.

Chapter 5

Gap infilling of stream flow records

Time series records of stream flow observations are essential for sustainable water management, since it constitutes the basis for all hydrological analyses (Brigode *et al.*, 2016). According to Campozano *et al.* (2014), they are especially useful in serving as indicators of past hydrological variability and are important contributors to hydrological models for predicting future stream flow behaviour (as seen in Chapter 4). The utility of such records for stream flow analyses is often dependent on continuous, uninterrupted observations. Interruptions in stream flow records (also referred to as gaps) are inevitable, especially for developing and economically emerging countries, and may serve as a serious drawback in the sustainable management of water resources (Campozano *et al.*, 2014). In this chapter, SVR and MLP models will be developed to address this problem. A review of some popular techniques currently used for infilling will be given in the following section, whereafter the main objectives and methodology for this part of the study will be defined.

5.1 Infilling techniques

Choosing an appropriate infilling technique for a given dataset has been investigated for decades and still remains a challenge. Three major classes of techniques can be distinguished: deterministic, stochastic and data-driven.

Deterministic methods comprise of mathematical functions that create continuous surfaces from measured data points, based on either the extent of similarity or the degree of smoothing. These interpolation techniques do not consider any randomness in the modelled system, and will therefore always produce the same output for a given initial state. Their robustness, ease of implementation and computational efficiency make them popular to use as infilling methods. The inverse distance weighting (IDW) method is one of the most frequently used deterministic approaches to approximate incomplete datasets in hydrology. The IDW method considers measured values of gaug-

ing stations surrounding the prediction location and makes the assumption that data from stations closer to the prediction location have a greater influence on the predictions than those farther away. This method may therefore be suitable if data from neighbouring stations on the same river channel are considered, but might not give desirable results for data from neighbouring stations on another watershed with a different surface, slope, permeability or overall morphology.

The radial basis function (RBF) is another popular deterministic interpolation technique and is one of the primary tools for interpolating multidimensional scattered data (Adhikary and Dash, 2017). It fits a surface through the measured data points and minimises the surface's total curvature (Wang *et al.*, 2014). As opposed to the IDW method, the RBF can make predictions with values above or below the maximum and minimum measured values, respectively. According to Adhikary and Dash (2017), the popularity of the RBF approach in environmental studies lies in its ability to handle arbitrarily scattered data and to generalise to several spatial dimensions.

Stochastic methods give probabilistic approximations of the modelled system's outcomes by utilising the statistical properties of the measured data points and quantifying the spatial autocorrelation and statistical relationships between them (Adhikary and Dash, 2017). Kriging is a popular stochastic interpolation technique for the infilling of climate time series records and has the ability to give unbiased predictions with minimum variance (Wang *et al.*, 2014). Ordinary and universal kriging are two of the most widely used methods and have been implemented for the interpolation or infilling of data in various hydrological studies (Campozano *et al.*, 2014). Stochastic methods are, however, computationally more expensive compared to deterministic techniques (Caldera *et al.*, 2016).

Data-driven techniques are also widely used for the infilling of incomplete climatic time series data, due to their ability to adapt to nonlinear relationships in the data. According to Campozano *et al.* (2014), neural networks and SVR approaches are especially popular for infilling in hydrological studies. For instance, Yozgatligil *et al.* (2013) applied an MLP type neural network, among other interpolation techniques, to infill meteorological time series data. Furthermore, Coulibaly and Evora (2007) investigated six different types of neural networks for the infilling of daily total precipitation records and daily extreme temperature series, of which the MLP method proved to be the most effective. Machine learning techniques have more recently been applied for the reconstruction of remote sensing observations of soil moisture and indicated improved results compared to more conventional models (Xing *et al.*, 2017).

5.2 Methodology

The main objective of this part of the study is to infill incomplete stream flow records using SVR and MLP models. A particular case will be addressed where two different gauging stations are located along a river channel: one with an uninterrupted, continuous stream flow record, and the other containing gaps. These two stations can also be referred to as the donor and target stations, respectively. Incomplete stream flow values of the target station will be infilled by considering the stream flow record of the donor station, as well as data from any rain gauges in the same catchment and in close proximity to the target station. The infilling models will be developed using the software described in Section 4.1.6.

5.2.1 Study area and data

High quality time series of daily stream flow and precipitation data were obtained from the Australian Bureau of Meteorology's Hydrologic Reference stations and Climate Data Online services. Two successive gauging stations in the Goulburn basin of Victoria were considered: station 405263 at Snake Creek Junction, and station 405219 downstream at Dohertys. These stations were considered as the donor and target stations, respectively. Data from rain gauge 083091 at Jamieson Licola, about 3.6 km away from the target station, were also considered. Figure 5.1 shows the location of these stations within the Goulburn basin.

The largest available sets of uninterrupted, coinciding rainfall and stream flow time series were considered, namely from 1 March 2004 to 14 December 2011. Two scenarios were investigated.

1. Firstly, we assumed that the donor station and rain gauge consisted of uninterrupted datasets, whereas the downstream target station had no available data for the whole of 2011. The training set then ranged from 1 March 2004 to 31 December 2010, and the test set from 1 January 2011 to 14 December 2011, as shown in Figure 5.2a. This specific configuration ensured that the models were evaluated based on their abilities to predict stream flow behaviour during any season of the year.
2. Secondly, we aimed to consider a more realistic representation of gaps in data records. The donor station and rain gauge were still assumed to have uninterrupted datasets, as was the case for the first scenario, but gaps of varying sizes were distributed throughout the dataset of the target station to capture different components of stream flow (including rising and falling limbs, peak flow and base flow). The training set for this scenario consisted of all data during which the target station had an

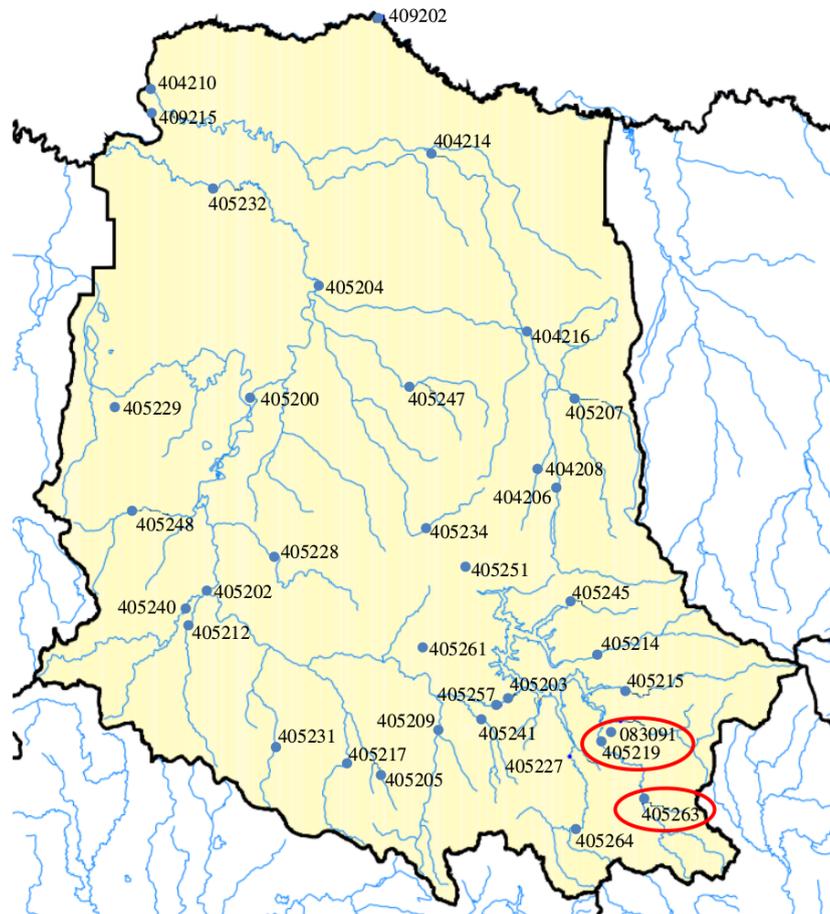
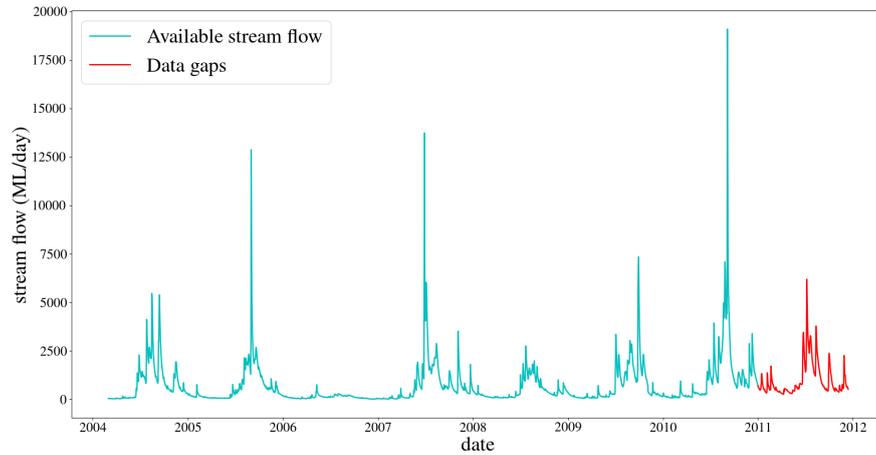


Figure 5.1: The Goulburn basin in Australia. The stations considered for this study are encircled.

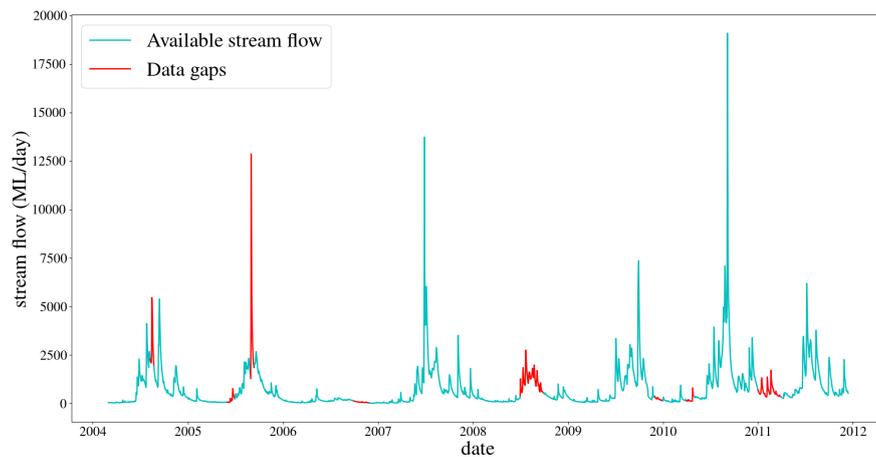
uninterrupted record, from 1 March 2004 to 14 December 2011, whereas the remaining data constituted the test set, as shown in Figure 5.2b.

Figure 5.3 shows the climatology graph of the average precipitation and stream flow data at the rain gauge and target station, respectively. It can be seen that precipitation is spread almost evenly throughout the year, whereas the average stream flow at the target station is lower during Summer and Autumn (December to May) and higher during Winter and Spring (June to November). It is therefore difficult to observe any relationship between precipitation and stream flow visually.

Figure 5.4 shows the climatology graph of the average stream flow data at the donor and target stations. A more visible relationship exists between the data of these two stations, since the behaviour of stream flow at one station closely corresponds to that of the other station. It is, however, difficult to establish the direct effect of the stream flow from the donor station on the stream flow at the target station.



(a)



(b)

Figure 5.2: Stream flow observations for the target station for (a) scenario 1 and (b) scenario 2. The cyan curves indicate available stream flow records used for training, whereas the red curves indicate data gaps that will be infilled using SVR and MLP.

5.2.2 Selection of input variables

A moving time window was considered for the generation of input and output data pairs, as was done in Chapter 4. For each measured stream flow value at the target station (which was considered as an output value), a corresponding input vector contained precipitation values from station 083091 and stream flow values from the donor station of the preceding p -day and u -day time windows, respectively. D refers to stream flow at the downstream target station, P represents precipitation, U refers to stream flow at the upstream donor station and t specifies the current day. An output value D_t then had an input vector $\{P_t, P_{t-1}, \dots, P_{t-p+1}, U_t, U_{t-1}, \dots, U_{t-u+1}\}$. No information from the target station was considered as possible input variables, to ensure that a

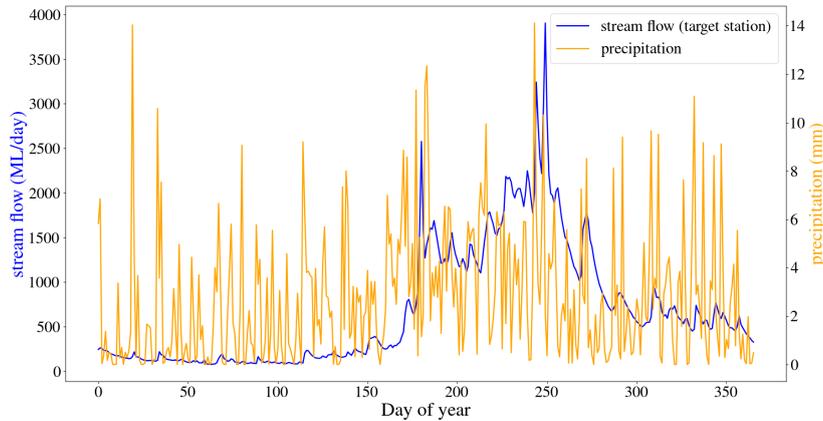


Figure 5.3: Stream flow and precipitation climatology graphs for the downstream target station 405219 and rain gauge 083091. No visible relationship exists between these two processes.

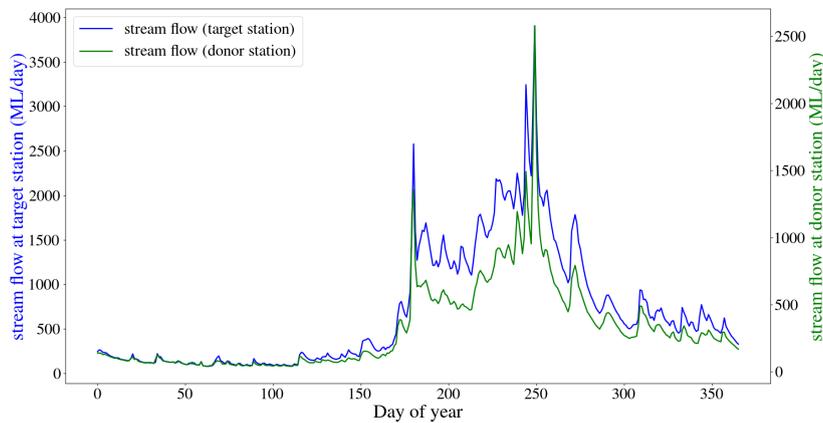


Figure 5.4: Stream flow climatology graphs for the target and donor stations, 405219 and 405263, respectively. A more visible relationship exists between the stream flow data at these two locations.

gap of any size in its dataset can be infilled.

The necessity of including precipitation in the input vector was also investigated by comparing the performance of the model to one that contained only stream flow, such that an output value D_t had input vector $\{U_t, U_{t-1}, \dots, U_{t-u+1}\}$. Furthermore, we investigated whether stream flow at the target station could be predicted by considering only antecedent precipitation values as input, such that an output value D_t had input vector $\{P_t, P_{t-1}, \dots, P_{t-p+1}\}$. We allowed the preceding p -day and u -day time windows to range from 1 to 7.

5.2.3 Preprocessing

Preprocessing was implemented in the same manner as in Chapter 4. The values in the dataset were linearly normalised to ensure that the influence of larger stream flow values would not dominate that of smaller precipitation values during training. Furthermore, 10-fold cross validation was considered for generalisation purposes.

5.2.4 Hyperparameters and network architecture

The SVR model with a radial basis kernel function, given by equation (3.36), was considered. As was the case in Section 4.1.4, an extensive grid search was performed to find the combination of parameters C , ϵ and γ with optimal performance during training, where C ranged from 1 to 10^4 , ϵ from 10^{-3} to 1 and γ from 10^{-4} to 1 (all on a logarithmic scale).

The MLP model structure consisted of preceding precipitation and stream flow values in the input layer, a single hidden layer, and a single stream flow prediction value in the output layer. Furthermore, the same method that was used to define bounds for the number of hidden nodes in Chapter 4 was also considered for this part of the study, namely a trial and error approach, with the number of hidden nodes ranging from $\log(n)$ to $2N+1$. The number of training samples are represented by n and the number of input nodes by N . An exhaustive search was performed to find the optimal activation function between the sigmoidal-type and logistic sigmoidal-type, given by equations (3.38) and (3.39) respectively, and the L-BFGS approach was considered for optimisation.

5.3 Results

Results of simulations for the optimal input features, hyperparameter combinations and model architecture for SVR and MLP models of the two scenarios discussed in Section 5.2.1 will be presented in the following subsections. The predictive capabilities of the machine learning models will also be evaluated, based on the efficiency criteria given in Section 3.1.4.

5.3.1 Feature selection

The preceding time windows for stream flow and precipitation that provided an optimal downstream infilling model for the target station were found using an exhaustive grid search. Since the SVR model contained a convex objective function, the optimal set of features and combination of hyperparameters provided a global minimum of the error function. However, since the MLP model considered a local optimisation algorithm that could get trapped in local min-

ima of the error function during training, 100 different random initializations were considered. The number of preceding precipitation and stream flow features resulting in the lowest average training error, together with the optimal number of hidden nodes and choice of activation function, were selected. Results are shown in Table 5.1.

It can be observed that the optimal preceding time windows for stream flow and precipitation of the SVR and MLP models varied. Precipitation seems to be an important input to the SVR models, since the maximum allowable number of antecedent precipitation values were selected for each scenario. When considering only precipitation as input, the MLP models also selected the maximum allowable number of features. However, when considering both precipitation and stream flow as input, the minimum number of allowable precipitation values were selected. It is noticeable that the optimal input feature sets of the SVR models were generally of a much higher dimension compared to that of the MLP models. Also, apart from the case where both stream flow and precipitation were considered as input to the SVR model, each model obtained the same number of optimal preceding stream flow and precipitation time windows for the two different scenarios.

A sensitivity analysis was done to determine whether the performance of the

Table 5.1: Optimal input features and hyperparameters in the SVR and MLP models for (a) scenario 1 and (b) scenario 2. C , ϵ and γ are SVR parameters, h is the number of nodes in the MLP hidden layer and g is the activation function. The model used precipitation data P from station 083091 for days $t - p + 1$ to t and stream flow data U from the donor station for days $t - u + 1$ to t to predict stream flow on day t at the target station. A dash sign signifies that no input from the particular process was considered as input.

(a) Scenario 1

Input vector	SVR					MLP			
	p	u	C	ϵ	γ	p	u	h	g
$\{P_t, P_{t-1}, \dots, P_{t-p+1}, U_t, U_{t-1}, \dots, U_{t-u+1}\}$	7	5	10	0.001	1	1	1	6	tanh
$\{U_t, U_{t-1}, \dots, U_{t-u+1}\}$	-	6	1	0.01	1	-	1	11	logistic
$\{P_t, P_{t-1}, \dots, P_{t-p+1}\}$	7	-	1000	0.01	0.001	7	-	12	tanh

(b) Scenario 2

Input vector	SVR					MLP			
	p	u	C	ϵ	γ	p	u	h	g
$\{P_t, P_{t-1}, \dots, P_{t-p+1}, U_t, U_{t-1}, \dots, U_{t-u+1}\}$	7	4	100	0.01	0.1	1	1	5	tanh
$\{U_t, U_{t-1}, \dots, U_{t-u+1}\}$	-	6	100	0.01	0.1	-	1	11	logistic
$\{P_t, P_{t-1}, \dots, P_{t-p+1}\}$	7	-	1000	0.01	0.01	7	-	13	logistic

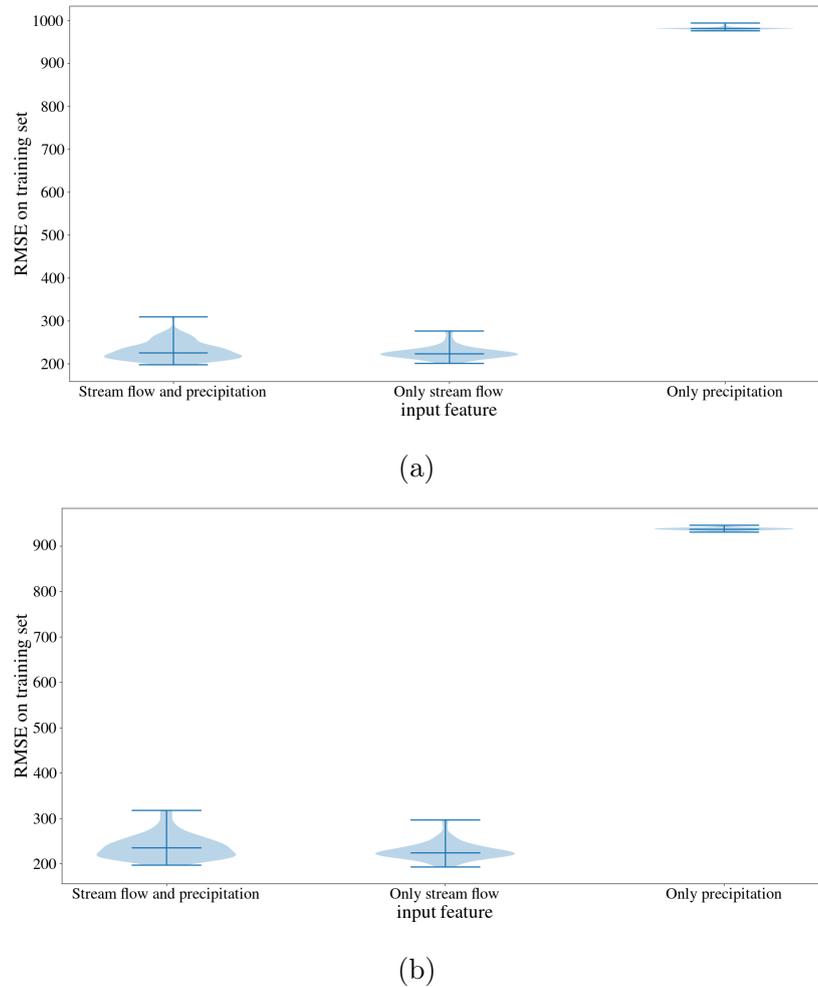


Figure 5.5: Violin plots showing the range and distribution of the training RMSE, of the optimal infilling models for (a) scenario 1 and (b) scenario 2.

MLP models were dependent on the initial weight vectors that were assigned to them. The RMSE on the training sets of the 100 models with different weight initializations are presented in the form of violin plots. A violin plot is similar to Tukey's box plots (Tukey, 1977), but adds additional information such as the sample data distribution and variations. Violin plots for the optimal gap infilling models listed in Table 5.1 are shown in Figure 5.5. For each model, the vertical line represents the range of the 100 RMSE values over the training set. The horizontal line between the two boundaries indicate the median value, and a kernel density estimation is given on each side of the vertical line to show the distribution shape of the data. It can be observed that the RMSE values for each model are concentrated at the median of the violin plots. Also, the range of the RMSE values are relatively small for each violin plot, indicating that the performance of the MLP models on the training sets were not considerably affected by the initial weight vectors.

5.3.2 Performance evaluation

The SVR and MLP models that performed optimally on the training and validation sets were applied to the test sets of the Goulburn stations. Pearson's correlation coefficient, the root mean squared error and the Nash-Sutcliffe efficiency were used to evaluate the performance of the models. Results are shown in Table 5.2. Three different kinds of models were investigated for the two infilling scenarios described in Section 5.2.1: one considering both antecedent stream flow and precipitation values as input, one containing only antecedent stream flow values, and one containing only antecedent precipitation values.

Scenario 1

The test set of scenario 1 ranged from 1 January 2011 to 14 December 2011 so that the machine learning infilling models' abilities to predict stream flow behaviour during different seasons of the year could be evaluated. As seen in Table 5.2a and Figure 5.6a, both the SVR and MLP models performed well on the given test set when considering a combination of antecedent precipitation and stream flow values as input. The SVR model slightly outperformed the MLP model and was able to approximate most of the peak flow values with high accuracy. However, both models significantly underestimated the last peak flow event in November 2011. When considering only antecedent stream flow values as input, a slight improvement in the performance of the MLP model was visible. However, as seen in Figure 5.6b, this input configuration

Table 5.2: Performance evaluation of our trained SVR and MLP models for (a) scenario 1 and (b) scenario 2 at the target station.

(a) Scenario 1

Input vector	r		RMSE		NSE	
	SVR	MLP	SVR	MLP	SVR	MLP
$\{P_t, P_{t-1}, \dots, P_{t-p+1}, U_t, U_{t-1}, \dots, U_{t-u+1}\}$	0.981	0.979	204	236	0.949	0.931
$\{U_t, U_{t-1}, \dots, U_{t-u+1}\}$	0.979	0.98	236	224	0.931	0.938
$\{P_t, P_{t-1}, \dots, P_{t-p+1}\}$	0.457	0.444	918	874	-0.029	0.066

(b) Scenario 2

Input vector	r		RMSE		NSE	
	SVR	MLP	SVR	MLP	SVR	MLP
$\{P_t, P_{t-1}, \dots, P_{t-p+1}, U_t, U_{t-1}, \dots, U_{t-u+1}\}$	0.991	0.994	247	232	0.956	0.961
$\{U_t, U_{t-1}, \dots, U_{t-u+1}\}$	0.991	0.988	231	273	0.961	0.946
$\{P_t, P_{t-1}, \dots, P_{t-p+1}\}$	0.220	0.323	1192	1172	-0.020	0.013

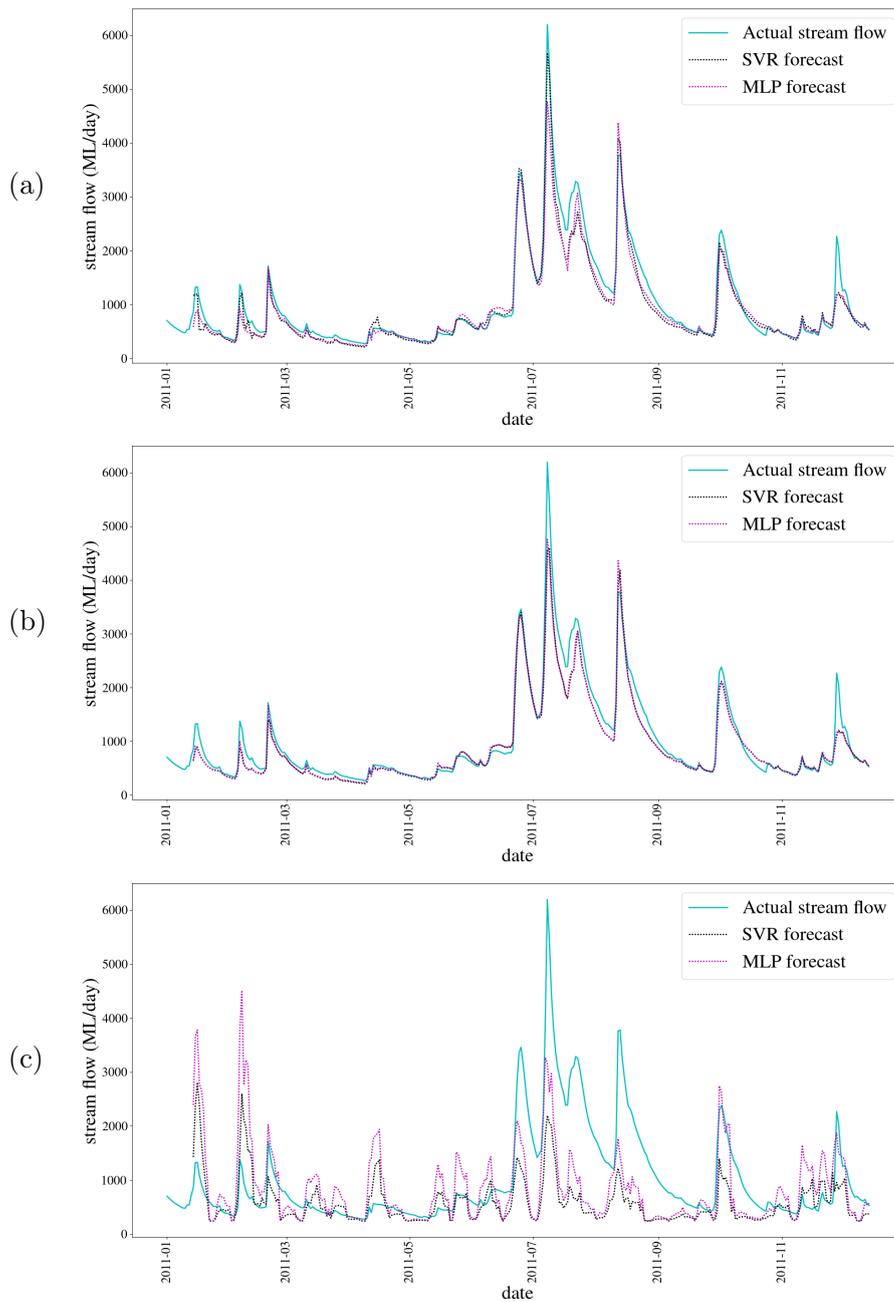


Figure 5.6: Gap infilling of the target station for scenario 1, considering (a) stream flow and precipitation, (b) only stream flow and (c) only precipitation as input to the model.

resulted in less accurate peak flow estimations when considering SVR.

Machine learning models that contained only preceding precipitation values as input were not able to accurately replicate stream flow behaviour at the target station. As seen in Table 5.2a, the MLP model obtained an NSE close to zero, and the SVR model obtained a negative NSE, which indicates that the mean value of the observed outcomes would have been a more reliable

predictor than the machine learning model itself. As seen in Figure 5.6c, the models had the ability to predict a rising or falling trend in stream flow, but could not determine the flow magnitude.

Scenario 2

The test set of scenario 2 consisted of varying sized gaps that were distributed throughout the dataset of the target station to give a more realistic representation of an interrupted stream flow record. An attempt to investigate different components of stream flow was performed by introducing gaps during different months each year. As seen in Table 5.2b, the models showed a performance similar to that of the first scenario in the sense that good results were obtained when considering either a combination of antecedent stream flow and precipitation values as input, or when considering only antecedent stream flow values. Only the figures showing infilling results for the former are given in this chapter.

Figure 5.7 shows the infilling results of the SVR and MLP models. A close-up of every segment is shown in Figure 5.8. It can be seen that both models were able to replicate the stream flow behaviour well. Even though the MLP model slightly outperformed the SVR model on average, no model outperformed the other on the infilling of every gap. For instance, as seen in Figure 5.8d, the SVR model was able to infill the stream flow gap from 1 October 2006 to 30 November 2006 well, whereas the MLP model predicted negative stream flow values. However, as seen in Figure 5.8h, the MLP model was able to predict the peak flow values with high accuracy, whereas the SVR model underestimated most of the peak flow events.

The SVR and MLP models that contained only preceding precipitation values as input were not able to accurately replicate the stream flow behaviour, as

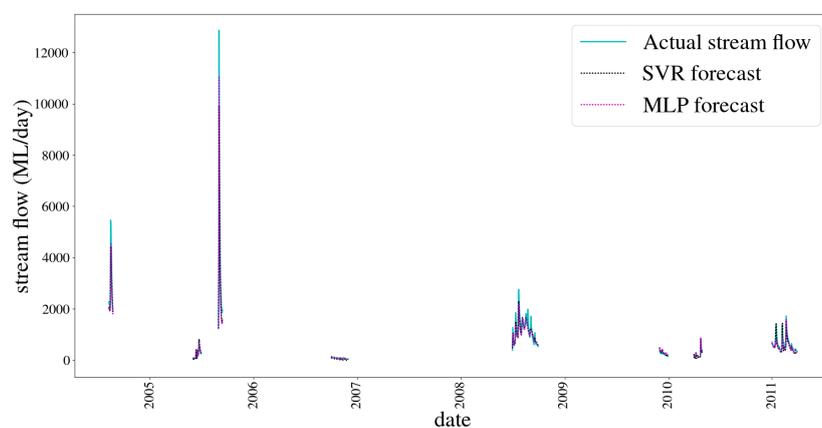


Figure 5.7: Gap infilling of station 405219 for scenario 2, considering antecedent stream flow and precipitation values as input. A close-up of every segment is shown in Figure 5.8.

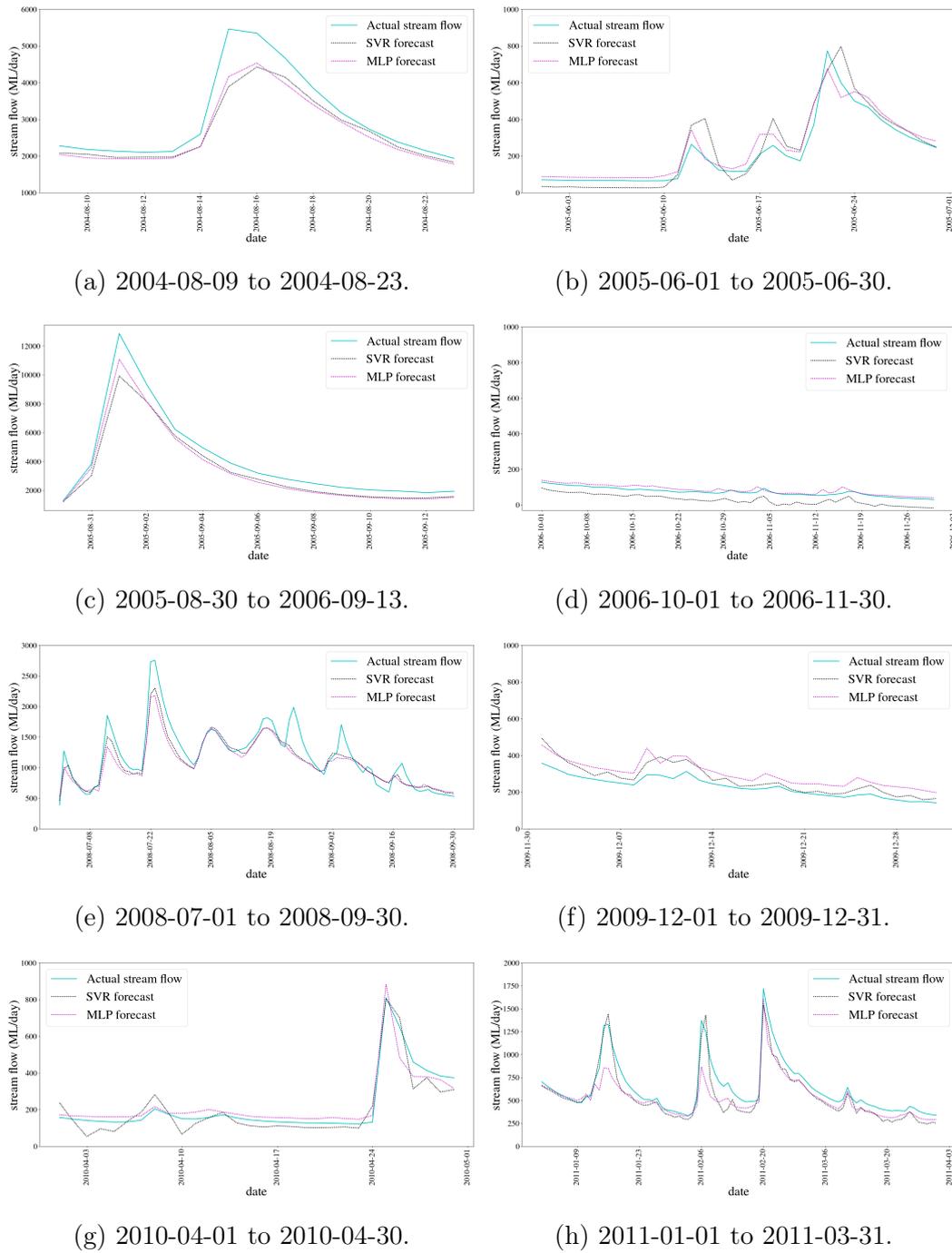


Figure 5.8: Various segments of the gap infilling of station 405219 for scenario 2, considering antecedent stream flow and precipitation values as input. Note the different vertical scales.

was the case in the first scenario. Both models obtained r and NSE values close to zero, indicating a weak linear correlation between the observed and predicted outputs, and implying that the machine learning models' predictive capabilities were equivalent to simply considering the mean observed output value as a predictor.

5.4 Discussion

Based on the results obtained for this part of the study, SVR and MLP models have the potential to be useful tools for the infilling of incomplete stream flow records, given that a donor station is located along the same river channel and contains a complete stream flow record over the infilling period. When only antecedent stream flow values from the donor station were considered as input, each machine learning technique was able to infill the target station's record with high accuracy and to model its stream flow behaviour properly. For this river site, it was sufficient to only consider stream flow data from the donor station without considering any antecedent precipitation data. The performance of the infilling model was therefore mainly dependent on the choice of donor station. What constitutes a good donor station might be a research topic in itself, but according to Harvey *et al.* (2010), standard considerations include the proximity and similarity (in terms of factors such as physiography and responsiveness) of the donor station to the target station. Furthermore, the use of multiple donor stations may strengthen the possibility of capturing more influences affecting flow at the target station.

Models that contained only antecedent stream flow values from the donor station as input misestimated some of the peak flow values. This could be attributed to the absence of important information about processes within the catchment area between the donor and target stations during training, leading to irreducible errors. For instance, rainfall events that occurred further downstream from the donor station may have affected the stream flow behaviour at the target station, but were not taken into account when developing the model. This may explain why the SVR model gave slightly better peak flow estimations when also considering preceding precipitation values as input.

The machine learning models were not able to predict stream flow at the target station by considering only antecedent precipitation values as input. Rising and falling trends were accurately predicted in some cases, but the flow magnitude could not be determined. This may be attributed to the fact that precipitation at only one location on the catchment was considered. As discussed in Chapter 4, rainfall events within close proximity to the considered gauging station, but at a different location than the considered rain gauge, may have affected the stream flow behaviour but were not taken into account when developing the model. Considering data from more rainfall stations within

close proximity to the target station may improve these results. Furthermore, as seen in Figure 5.1, the rain gauge is downstream from the target station. Since it is only 3.6 km away, an assumption was made that the rainfall events recorded at this gauge would also reach the stream at the target station. However, no information about the distribution or movement of the rainfall events was given. The rainfall recorded at station 083091 may therefore not even have reached the stream at the target station. Therefore, considering rain gauges upstream from the target station may improve results.

As already mentioned, a donor station and nearby rain gauge with complete, uninterrupted records over the infilling period were considered for this part of the study. However, uninterrupted records may not always be available and may limit the infilling capabilities of our models. For instance, consider a gap in the target station's record that has to be infilled using the SVR approach, and assume that only the previous 4 stream flow values and 3 precipitation values from the donor and rainfall stations, respectively, are available. The optimal SVR model requires 7 antecedent precipitation values and 5 antecedent stream flow values and can therefore not be used to infill this gap. However, to overcome this limitation, a model with an appropriate input feature vector can be trained. This can be done by performing an extensive grid search to find the preceding p -day and u -day time windows, ranging from 1 to 3 and 1 to 4, respectively, that will provide an optimal infilling model for this specific scenario. Another option is to construct a forecasting model such as those described in Chapter 4 to estimate the missing values at the donor station. This could be especially useful if the stream flow records at the donor and target stations contain gaps on the same day.

A limitation of this infilling methodology could be the unavailability of donor stations. Many rivers do not necessarily contain successive gauging stations on the same channel or tributary. Furthermore, existing donor stations may be too far from the target station, causing large irreducible errors during training and preventing the machine learning models from adequately replicating stream flow behaviour at the target station. The importance of the geographical locations of donor stations relative to target stations were, however, not investigated, and may be examined in future research.

Chapter 6

Conclusions and recommendations

This study investigated the ability of machine learning models to overcome certain challenges faced within the hydrological domain concerning short-term stream flow forecasting and gap infilling. For the first part of the study, SVR and MLP models were employed to forecast stream flow at the Shoalhaven, Herbert and Adelaide gauging stations with a lead time of up to 7 days. The predictive capabilities of these machine learning models were compared to that of a physically based rainfall-runoff model, provided by the Australian Bureau of Meteorology. For 1 day lead time forecasts, each machine learning model properly modelled the stream hydrograph shape and the times to peak. However, a noticeable decrease in predictive capabilities with an increase in lead time occurred, which could be attributed to the absence of important information about catchment processes, leading to irreducible errors in the model. Based on the evaluation criteria, both the MLP and SVR models performed better than the BOM model for the Shoalhaven station. For the other stations, no single model outperformed the others.

Based on the results obtained for this part of the study, SVR and MLP models have the potential to be useful tools for short-term stream flow forecasting. They do not require specialized knowledge of physical phenomena, and are therefore especially useful when it is difficult to build a physically based model due to a lack of complete understanding of the underlying processes. This could be seen particularly in the results of the Shoalhaven river site, where the BOM model significantly overpredicted the stream flow values due to its incorrect estimations of precipitation. Moreover, it can be concluded that machine learning models may be helpful to use as modelling alternatives and to validate results obtained from physically based models. They are also computationally efficient in the sense that once they are trained, predictions can be made fairly quickly. Machine learning models could also be combined with physically based models for more accurate hybrid forecasting techniques, a topic that may be considered for future research.

A limitation of machine learning models is, however, that a sufficient amount of

historical stream flow and precipitation data records should be available. Many of the existing gauging stations have datasets of limited size, or a considerable number of gaps. For the second part of the study we addressed this problem using SVR and MLP as infilling techniques. A particular case was considered where two different gauging stations were located along a river channel: a donor station with an uninterrupted, continuous stream flow record, and a target station containing gaps. Two neighbouring gauging stations and a rain gauge on the Goulburn river station were considered for experimentation. The incomplete stream flow record of the target station were infilled using data from the donor station and rain gauge.

The results indicated a promising role of machine learning applications for the infilling of gaps in stream flow records. Based on performance analyses, results with high accuracy were obtained for both the SVR and MLP models by merely considering preceding stream flow values from the donor station as input to the models, without considering any data from rain gauges. Incorporating precipitation in the SVR model provided slightly better peak flow estimations, but did not show significant improvements in the overall prediction accuracy of the models. Constructing an input vector that contains data only from donor stations and not from the target station is advantageous in the sense that the model performance is not affected by the size of the gap that has to be infilled.

Proposed future work entails the investigation of issues such as the influence of donor station choice and the potential for infilling approaches to perform differently when considering varying flow magnitudes or regimes. A wider range of machine learning infilling techniques could also be examined to determine discrepancies and similarities between model performances for certain catchment characteristics, to enhance the development of infilling practices. This research could potentially assist in systematically infilling gaps in stream flow records to improve the utility of flow data to end users.

References

- Adhikari, R. and Agrawal, R. (2013). *An Introductory Study on Time Series Modeling and Forecasting*. Lambert Academic Publishing.
- Adhikary, P. and Dash, C. (2017). Comparison of deterministic and stochastic methods to predict spatial variation of groundwater depth. *Applied Water Science*, vol. 7, no. 1, pp. 339–348.
- Belayneh, A. and Adamowski, J. (2013). Drought forecasting using new machine learning methods. *Journal of Water and Land Development*, vol. 18, pp. 3–12.
- Berry, M. and Linoff, G. (1997). *Data Mining Techniques*. John Wiley & Sons.
- Bhagwat, P. and Maity, R. (2012). Multistep-ahead river flow prediction using LS-SVR at daily scale. *Journal of Water Resource and Protection*, vol. 4, pp. 528–539.
- Borji, M., Malekian, A., Salajegheh, A. and Ghadimi, M. (2016). Multi-time-scale analysis of hydrological drought forecasting using support vector regression (SVR) and artificial neural networks (ANN). *Arabian Journal of Geosciences*, vol. 9, pp. 1–10 (article no. 725).
- Bowden, G., Dandy, G. and Maier, H. (2003). Data transformation for neural network models in water resources applications. *Journal of Hydroinformatics*, vol. 5, no. 4, pp. 245–258.
- Bowden, G., Dandy, G. and Maier, H. (2005). Input determination for neural network models in water resources applications. Part 1: background and methodology. *Journal of Hydrology*, vol. 301, no. 1-4, pp. 75–92.
- Box, G. and Jenkins, G. (1970). *Time-series Analysis: Forecasting and Control*. Holden-Day.
- Bradley, S., Hax, A. and Magnanti, T. (1977). *Applied Mathematical Programming*. Addison-Wesley.
- Bray, M. and Han, D. (2004). Identification of support vector machines for runoff modelling. *Journal of Hydroinformatics*, vol. 6, no. 4, pp. 265–280.
- Brigode, P., Brissette, F., Nicault, A., Perreault, L., Kuentz, A. Mathevet, T. and Gailhard, J. (2016). Streamflow variability over the 1881-2011 period in northern Québec: comparison of hydrological reconstructions based on tree rings and geopotential height field reanalysis. *Climate of the Past*, vol. 12, pp. 1785–1804.
- Brown, R. (2012). *Optimization and Inductive Models for Continuous Estimation of Hydrologic Variables*. Master's Thesis, Florida Atlantic University.

- Bureau of Meteorology (2017). 7-day Streamflow Forecasts.
Available at: <http://www.bom.gov.au/water/7daystreamflow/about.shtml>
- Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2, pp. 121–167.
- Caldera, H., Piyathisse, V. and Nandalal, K. (2016). A comparison of methods estimating missing daily rainfall data. *Engineer: Journal of the Institution of Engineers*, vol. 49, no. 4, pp. 1–8.
- Campozano, L., Sanchez, E. and Samaniego, E. (2014). Evaluation of infilling methods for time series of daily precipitation and temperature: the case of the Ecuadorian Andes. *Maskana*, vol. 5, no. 1, pp. 99–115.
- Chang, F., Chiang, Y. and Chang, L. (2007). Multi-step-ahead neural networks for flood forecasting. *Hydrological Sciences Journal*, vol. 52, no. 1, pp. 114–130.
- Coulibaly, P. and Evora, N. (2007). Comparison of neural network methods for infilling missing daily weather records. *Journal of Hydrology*, vol. 341, no. 1-2, pp. 27–41.
- De Vos, N. and Rientjies, T. (2005). Constraints of artificial neural networks for rainfall-runoff modeling: trade-offs in hydrological state representation and model evaluation. *Hydrology and Earth System Sciences Discussions*, vol. 9, pp. 111–126.
- Dibike, Y., Velickov, S., Solomatine, D. and Abbott, M. (2001). Model induction with support vector machines: introduction and application. *Journal of Computing in Civil Engineering*, vol. 15, pp. 208–216.
- Encyclopaedia Britannica (2017). Water cycle. [Online; accessed September, 2017].
Available at: <https://www.britannica.com/science/water-cycle>
- Falconer, R., Lin, B. and Harpin, R. (2005). Environmental modelling in river basin management. *International Journal of River Basin Management*, vol. 3, no. 1, pp. 169–184.
- Ghana Sheila, K. and Deepak, S. (2013). Review on methods to fix number of hidden neurons in neural networks. *Mathematical Problems in Engineering*, vol. 2013.
- Golden, R. (1996). *Mathematical Methods for Neural Network Analysis and Design*. MIT Press.
- Granata, F., Gargano, R. and De Marinis, G. (2016). Support vector regression for rainfall-runoff modeling in urban drainage: a comparison with the EPA’s storm water management model. *Water*, vol. 8, no. 3, pp. 1–13 (article no. 69).
- Harvey, C., Dixon, H. and Hannaford, J. (2010). Developing best practice for infilling daily river flow data. In: *Proceedings of the 3rd International Symposium of the British Hydrological Society*, pp. 1–8.
- Hastie, T., Tibshirani, R. and Friedman, J. (2009). Model assessment and selection. In: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd Edition*, chap. 7, pp. 241–249. Springer-Verlag.
- Imrie, C., Durucan, S. and Korre, A. (2000). River flow prediction using artificial neural networks: generalisation beyond the calibration range. *Journal of Hydrology*, vol. 233, pp. 138–153.

- Karsoliya, S. (2012). Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture. *International Journal of Engineering Trends and Technology*, vol. 3, no. 6, pp. 714–717.
- Karush, W. (1939). *Minima of Functions of Several Variables with Inequalities as Side Constraints*. Master's Thesis, University of Chicago.
- Khotanzad, A., Afkhami-Rohani, R., Lu, T., Abaye, A., Davis, M. and Maratukulamm, D. (1997). ANNSTLF – a neural-network-based electric load forecasting system. *IEEE Transactions on Neural Networks*, vol. 8, no. 4, pp. 835–846.
- Knapp, H., Durgunoglu, A. and Ortel, T. (1991). *A review of rainfall-runoff modeling for stormwater management*. Illinois State Water Survey.
- Kokkonen, T. and Jakeman, A. (2001). A comparison of metric and conceptual approaches in rainfall-runoff modeling and its implications. *Water Resources Research*, vol. 37, no. 9, pp. 2345–2352.
- Krause, P., Boyle, D. and Bäse, F. (2005). Comparison of different efficiency criteria for hydrological model assessment. *Advances in Geosciences*, vol. 5, pp. 89–97.
- Krishna, B. (2014). Comparison of wavelet-based ANN and regression models for reservoir inflow forecasting. *Journal of Hydrologic Engineering*, vol. 19, pp. 1385–1400.
- Kuhn, H. and Tucker, A. (1951). Nonlinear programming. In: *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probabilistics*, pp. 299–324.
- Lee, Y., Yeh, Y. and Pao, H. (2012). *Introduction to Support Vector Machines and their Applications in Bankruptcy Prognosis*, chap. 7, pp. 731–761. Springer.
- Linsey, R., Kohler, M. and Paulhus, J. (1949). *Applied Hydrology*. McGraw-Hill.
- Maier, H. and Dandy, G. (2000). Neural networks for the prediction and forecasting of water resources variables: a review of modeling issues and applications. *Environmental Modelling & Software*, vol. 15, pp. 101–124.
- Maier, H., Dandy, G. and Sudheer, K. (2010). Methods used for the development of neural networks for the prediction of water resource variables in river systems: current status and future directions. *Environmental Modelling & Software*, vol. 25, no. 8, pp. 891–909.
- Masters, T. (1993). *Practical Neural Network Recipes in C++*. Academic Press.
- Mehr, A., Kahya, E., Sahin, A. and Nazemosadat, M. (2015). Successive-station monthly streamflow prediction using different artificial neural network algorithms. *International Journal of Environmental Science and Technology*, vol. 12, pp. 2191–2200.
- Mishra, A. and Desai, V. (2006). Drought forecasting using feed-forward recursive neural network. *Ecological Modelling*, vol. 198, no. 1–2, pp. 127–138.
- Nezhad, M., Zhu, D., Li, X., Yang, K. and Levy, P. (2016). SAFS: A deep feature selection approach for precision medicine. In: *IEEE International Conference on Bioinformatics and Biomedicine*, pp. 501–506.
- Noori, N. and Kalin, L. (2016). Coupling SWAT and ANN models for enhanced daily streamflow. *Journal of Hydrology*, vol. 533, pp. 141–151.

- Oyebode, O. (2014). *Modelling Streamflow Response to Hydro-climatic Variables in the Upper Mkomazi River, South Africa*. Master's thesis, Durban University of Technology.
- Panchal, F. and Panchal, M. (2014). Review on methods of selecting number of hidden nodes in artificial neural network. *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 11, pp. 455–464.
- Parisi, R., Di Claudio, E., Orlandi, G. and Rao, B. (1996). A generalized learning paradigm exploiting the structure of feedforward neural networks. *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1451–1460.
- Perrin, C., Michel, C. and Andréassian, V. (2003). Improvement of a parsimonious model for streamflow simulation. *Journal of Hydrology*, vol. 279, pp. 275–289.
- Raghavendra, S. and Deka, P. (2014). Support vector machine applications in the field of hydrology: a review. *Applied Soft Computing*, vol. 19, pp. 372–286.
- Remesan, R. and Mathew, J. (2014). *Hydrological Data Driven Modelling: a Case Study Approach*. Springer.
- Roy, A., Dybas, A., Fritz, K. and Lubbers, H. (2009). Urbanization affects the extent and hydrologic permanence of headwater streams in a midwestern US metropolitan area. *Journal of the North American Benthological Society*, vol. 28, no. 4, pp. 911–928.
- Saghafian, B. and Julien, P. (1995). Time to equilibrium for spatially variable watersheds. *Journal of Hydrology*, vol. 172, pp. 231–245.
- Salavati, B., Oudin, L., C., F.-P. and Ribstein, P. (2016). Modeling approaches to detect land-use changes: urbanization analyzed on a set of 43 US catchments. *Journal of Hydrology*, vol. 538, pp. 138–151.
- Shanno, D. (1978). Conjugate gradient methods with inexact line searches. *Mathematics of Operations Research*, vol. 3, pp. 244–256.
- Shi, J. (2000). Reducing prediction error by transforming input data for neural networks. *Journal of Computing in Civil Engineering*, vol. 14, no. 2, pp. 109–116.
- Sinclair Knight Merz (2010). Developing guidelines for the selection of stream-flow gauging stations. Final report, prepared for the climate and water division, Bureau of Meteorology.
- Skajaa, A. (2010). *Limited Memory BFGS for Nonsmooth Optimization*. Master's thesis, New York University.
- Solomatine, D. and Ostfeld, A. (2008). Data-driven modeling: some past experiences and new approaches. *Journal of Hydroinformatics*, vol. 10, no. 1, pp. 3–22.
- Solomatine, D., See, L. and Abrahart, R. (2008). *Data-Driven Modelling: Concepts, Approaches and Experiences*, pp. 17–30. Springer Berlin Heidelberg.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958.
- Subramanya, K. (2009). *Engineering Hydrology: 3rd Edition*. McGraw-Hill.

- Taver, V., Johannet, A., Borrell-Estupina, V. and Pistre, S. (2015). Feedforward vs recurrent neural network models for non-stationarity modelling using data assimilation and adaptivity. *Hydrological Sciences Journal*, vol. 60, no. 7–8, pp. 1242–1265.
- Tencaliec, P., Favre, A., Prieur, C. and Mathevet, T. (2015). Reconstruction of missing daily streamflow data using dynamic regression models. *Water Resources Research*, vol. 51, no. 12, pp. 9447–9463.
- Thissen, U., Van Brakel, R., De Weijer, A., Melssen, W. and Buydens, L. (2003). Using support vector machines for time series prediction. *Chemometrics and Intelligent Laboratory Systems*, vol. 69, pp. 35–69.
- Tokar, A. and Johnson, P. (1999). Rainfall-runoff modeling using artificial neural networks. *Journal of Hydrologic Engineering*, vol. 4, no. 3, pp. 232–239.
- Tukey, J. (1977). *Exploratory Data Analysis*. Addison-Wesley.
- Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer.
- Wanas, N., Auda, G., Kamel, M. and Karray, F. (1998). On the optimal number of hidden nodes in a neural network. In: *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 918–921.
- Wang, S., Huang, G., Lin, Q., Li, Z., Zhang, H. and Fan, Y. (2014). Comparison of interpolation methods for estimating spatial distribution of precipitation in Ontario Canada. *International Journal of Climatology*, vol. 34, pp. 3745–3751.
- Wang, Y., Guo, S., Xiong, L., Liu, P. and Liu, D. (2015). Daily runoff model based on ANN and data preprocessing techniques. *Water*, vol. 7, pp. 4144–4160.
- Wilson, E.M. (1974). *Hydrograph Analysis*, pp. 120–153. Macmillan Education UK.
- Wisler, C. and Brater, E. (1959). *Hydrology: 2nd Edition*. John Wiley & Sons.
- Xing, C., Chen, N., Zhang, X. and Gong, J. (2017). A machine learning based reconstruction method for satellite remote sensing of soil moisture images with in situ observations. *Remote Sensing*, vol. 9, no. 5, pp. 1–24 (article no. 484).
- Yaseen, Z., El-shafie, A., Jaafar, O. and H.A., A. (2015). Artificial intelligence based models for stream-flow forecasting: 2000-2015. *Journal of Hydrology*, vol. 530, pp. 829–844.
- Yozgatligil, C., Aslan, S., Iyigun, C. and Batmaz, I. (2013). Comparison of missing value imputation methods in time series: the case of Turkish meteorological data. *Theoretical and Applied Climatology*, vol. 112, pp. 143–167.
- Zealand, C., Burn, D. and Simonovic, S. (1999). Short term streamflow forecasting using artificial neural networks. *Journal of Hydrology*, vol. 214, pp. 32–48.
- Zhang, X., Amirthanathan, G., Bari, M., Laugesen, R., Shin, D., Kent, D., MacDonald, A., Turner, M. and Tuteja, N. (2016). How streamflow has changed across Australia since the 1950s: evidence from the network of hydrologic reference stations. *Hydrology and Earth System Sciences*, vol. 20, pp. 3947–3965.