Use of Artificial Neural Networks for Modelling Multivariate Water Quality Time Series

By

Holger Robert Maier
B.E. (Hons)

Thesis submitted for the degree of
Doctor of Philosophy
in
The University of Adelaide
(Faculty of Engineering)

November 1995

To my loving parents:

Wilhelm and Erika Maier

Contents

List of Figures ..... **ix**

List of Tables ..... **xx**

Abstract ..... **xxviii**

Statement of Originality ..... **xxix**

Acknowledgments ..... **xxx**

1. Introduction **1**

1.1 Introduction 1

1.2 Objectives and Scope 2

1.3 Layout and Contents of Thesis 3

2. Literature Review **5**

2.1 Artificial Neural Networks 5

2.1.1 Introduction 5

2.1.2 Artificial Neural Networks in Perspective 6

2.1.2.1 Parallel Versus Serial Computing 6

2.1.2.2 ANNs as a Form of Artificial Intelligence 6

2.1.3 History of Neural Computing 8

2.1.4 Natural Neural Networks 11

2.1.5 Artificial Neural Networks 12

2.1.5.1 Introduction 12

2.1.5.2 Operation 13

2.1.5.3 Architecture 21

2.1.5.4 Performance 21

2.1.6 Implementation of Artificial Neural Networks 21
2.1.6.1 Software Simulation 22
2.1.6.2 Hardware Accelerators 22
2.1.6.3 Optical Processors 22
2.1.6.4 Direct Hardware Implementations 22
2.1.7 Different Neural Network Architectures 23
2.1.7.1 Network Classification 23
2.1.8 Back-Propagation Networks 27
2.1.8.1 Learning 29
2.1.8.2 Measuring Network Performance 35
2.1.8.3 Network Design 36
2.1.8.4 Deficiencies of the Back-Propagation Algorithm 41
2.1.8.5 Ways of Improving the Back-Propagation Algorithm 42
2.1.9 Applications 51
2.1.9.1 Networks for Pattern Recognition 52
2.1.9.2 Networks for Conceptualisation 53
2.1.9.3 Networks for Filtering 53
2.1.9.4 Networks for Optimisation 53
2.1.9.5 Networks for Prediction 55
2.2 Time Series Analysis 63
2.2.1 Introduction 63
2.2.2 Time Series Models 63
2.2.3 Linearity 65
2.2.4 Distribution of Data 65
2.2.5 Spacing of Observations 66
2.2.6 Stationarity 66
2.2.6.1 Plotting the Data 67
2.2.6.2 The Autocorrelation Function 67
2.2.6.3 The Partial Autocorrelation Function 69
2.2.7 Transformations 71
2.2.7.1 The Classical Decomposition Model 71
2.2.7.2 Differencing 72
2.2.8 Model Identification 74
2.2.8.1 Non-Seasonal Models 75
2.2.8.2 Seasonal Models 78
2.2.8.3 Akaike's Information Criterion (AIC) 82
2.2.9 Parameter Estimation 83
2.2.10 Diagnostic Checking 84
2.2.10.1 The Durbin - Watson Statistic 85

2.2.10.2 The Box - Pierce Method 85
2.2.10.3 The Portmanteau Lack-of-fit Test 86
2.2.11 Forecasting 87
2.2.12 The Box-Jenkins Methodology 87
2.2.13 Multivariate Time Series Models 89
2.2.13.1 Introduction 89
2.2.13.2 Classes of Multivariate Time Series Models 90
2.2.13.3 The Cross-Correlation Function 93
2.2.13.4 Stationarity 95
2.2.13.5 Model Identification 96
2.2.13.6 Parameter Estimation 97
2.2.13.7 Diagnostic Checking 99
2.2.13.8 State Space Models 100
2.2.13.9 The Kalman Filter 105
2.2.13.10 Time Variable Parameter (TVP) Models 107
2.2.13.11 Forecasting 108
2.2.13.12 Applications 109
2.3 Artificial Neural Networks for Modelling Time Series 110
2.3.1 Inspection of Plots of the Time Series 111
2.3.2 Determination of the Model Inputs 111
2.3.3 Choice of the Network Parameters and Geometry 113
2.3.4 Training of the Network 113
2.3.5 Validation of the Network 114
2.3.6 Forecasting 114
2.3.7 Other Differences Between ANN and ARMA Type Models 115

3. Salinity Forecasting in the River Murray 116
3.1 Introduction 116
3.2 Background 117
3.2.1 The Salinity Problem in the River Murray 123
3.2.1.1 Factors Causing Salinity 124
3.2.1.2 Factors Increasing Saline Accessions 126
3.2.1.3 Effects of High Salinity 127
3.2.1.4 Remedial and Preventive Measures 131
3.2.2 Case Study 132
3.3 Salinity Forecasting Models 133
3.3.1 Forecasting Methods 133
3.3.2 Forecasting Period 134
3.3.3 Available Data 134

3.3.3.1 Inspection of Plots of Data 136
3.3.4 Implementation 161
3.3.4.1 ANN Models 161
3.3.4.2 Time Series Models 162
3.3.5 Performance Measures 164
3.3.6 Nomenclature 165
3.4 Development of Univariate Neural Network Model 166
3.4.1 Introduction 166
3.4.2 Checking of the Time Series for Stationarity 167
3.4.3 Transformation of the Data by Differencing 169
3.4.4 Development of Models Using the Differenced Data 174
3.4.4.1 Determination of Initial Model Inputs 175
3.4.4.2 One Day Forecasts 177
3.4.4.3 Fourteen Day Forecasts 181
3.4.4.4 Summary 183
3.4.5 Development of Models Using the Raw Data 183
3.4.5.1 Determination of Initial Model Inputs 184
3.4.5.2 One Day Forecasts 184
3.4.5.3 Fourteen Day Forecasts 189
3.4.5.4 Summary 199
3.4.6 Results / Discussion 199
3.4.7 Real Time Forecasts 201
3.5 Development of Multivariate Neural Network Model 203
3.5.1 Introduction 203
3.5.2 Determination of Model Inputs 203
3.5.2.1 Determination of Initial Model Inputs 204
3.5.2.2 Choice of Internal Parameters and Network Geometry 207
3.5.2.3 Training / Testing 208
3.5.2.4 Sensitivity Analysis 210
3.5.2.5 Effect of the Number of Outputs 213
3.5.2.6 Results / Discussion 214
3.5.3 Investigation of the Effects of Internal Parameters and Network Geometry 217
3.5.3.1 Introduction 217
3.5.3.2 Method 218
3.5.3.3 Effect of Internal Parameters on Network Performance 219
3.5.3.4 Effect of Geometry on Network Performance 250

3.5.3.5 Application of Results 255
3.5.3.6 Generalisation ability at various stages of learning 258
3.5.3.7 Conclusions 279
3.5.4 Real Time Forecasts 284
3.5.5 Summary / Conclusions 289
3.6 Development of Univariate Time Series Model 290
3.6.1 Introduction 290
3.6.2 Inspection of the Plot of the Time Series 291
3.6.3 Checking of the Time Series for Stationarity 291
3.6.4 Transformation of Data by Differencing 291
3.6.5 Identification of Model Type and Order 293
3.6.6 Estimation of Model Parameters 295
3.6.7 Diagnostic Checking 299
3.6.8 Forecasting 301
3.6.9 Results / Discussion 302
3.6.10 Summary / Conclusions 312
3.7 Development of Multivariate Time Series Model 313
3.7.1 Introduction 313
3.7.2 Choice of Component Time Series 313
3.7.3 Inspection of Plots of the Time Series 313
3.7.4 Checking of the Time Series for Joint Stationarity 314
3.7.5 Transformation of the Time Series by Differencing 315
3.7.6 Identification of the Class and Initial Order of the Model Predicting Salinity at Murray Bridge 318
3.7.6.1 Model Class 318
3.7.6.2 Model Order 319
3.7.7 Identification of the Initial Order of the Models for the Time Series Required for the Prediction of Salinity at Murray Bridge 329
3.7.8 Formulation of the Model in State Space Form 333
3.7.9 Parameter Estimation and Forecasting 338
3.7.10 Results / Discussion 350
3.7.11 Diagnostic Checking 356
3.7.12 Summary / Conclusions 359
3.8 Comparison of Results 360
3.8.1 Introduction 360
3.8.2 Artificial Neural Network Models 360
3.8.3 Time Series Models 361
3.8.4 Univariate Models 363
3.8.5 Multivariate Models 365

3.8.6 Artificial Neural Network Versus ARMA Type Models 367
3.9 Longer Term Salinity Forecasts 368
3.9.1 Introduction 368
3.9.2 Determination of Model Inputs 368
3.9.3 Choice of Internal Parameters and Network Geometry 369
3.9.4 Investigation of the Generalisation Ability of the Models 369
3.9.4.1 Sensitivity of Models to Flow Inputs 373
3.9.5 Real Time Forecasting 376
3.10 Methods for Determining Inputs to Multivariate Neural Network Models 380
3.10.1 Determination of Model Inputs Using the Method of Haugh and Box (Method 1) 381
3.10.2 Determination of Model Inputs Using the Neural Network Based Approach (Method 2) 382
3.10.2.1 Inspection of Plots of the Time Series 382
3.10.2.2 Development of Univariate and Bivariate ANN Models 382
3.10.2.3 Performance of Sensitivity Analyses 387
3.10.3 Summary of Network Inputs Chosen 391
3.10.4 Real Time Forecasting 393
3.10.5 Results / Discussion 394
3.11 Conclusions 396

4. Modelling Blue-Green Algae in the River Murray 402
4.1 Introduction 402
4.2 Background 403
4.2.1 Effects of Cyanobacterial Blooms 405
4.2.1.1 Public Health Effects 405
4.2.1.2 Effects on Animal Health 410
4.2.1.3 Ecological Effects 410
4.2.1.4 Aesthetic Effects 411
4.2.1.5 Effects on Water Supply Operations 412
4.2.1.6 Effects on Recreation and Tourism 412
4.2.2 Addressing the Cyanobacteria Problem 412
4.2.2.1 Water Treatment 412
4.2.2.2 Prevention of Algal Blooms 416
4.2.3 Modelling the Incidence of Algal Populations 420
4.3 Case Study 422
4.3.1 Available Data 423

4.3.1.1 Inspection of Plots of Data 425
4.3.2 Implementation 431
4.3.3 Forecasting Period 431
4.3.4 Performance Measures 431
4.3.5 Nomenclature 432
4.4 Development of Multivariate Neural Network Model 433
4.4.1 Inspection of Plots of the Time Series 433
4.4.2 Determination of Model Inputs 433
4.4.2.1 Development of Bivariate Models 433
4.4.2.2 Performance of Sensitivity Analyses 441
4.4.3 Choice of Network Geometry and Internal Parameters 446
4.4.4 Training 446
4.4.5 Results / Discussion 447
4.5 Conclusions 453

5. Conclusions and Recommendations 454
5.1 Contributions of the Research 454
5.2 General Conclusions 456
5.3 Specific Conclusions 462
5.3.1 Case Study 1 - Forecasting Salinity in the River Murray 462
5.3.2 Case Study 2 - Modelling Blue-Green Algae in the River Murray 462
5.4 Recommendations for Further Work 463

A Notation 465

B Abbreviations 470

C Training / Testing Sets 473
C.1 Forecasting Salinity in the River Murray 473
C.2 Modelling Blue-Green Algae in the River Murray 479

D Neural Network Parameters and Geometries 482
D.1 Forecasting Salinity in the River Murray 482
D.2 Modelling Blue-Green Algae in the River Murray 492

E Neural Network Performance at Various Stages of Learning (Test Set) 493
E.1 One Day Forecasts 493
E.2 14 Day Forecasts 495
E.3 28 Day Forecasts 522

Bibliography 526

List of Figures

2.1 Typical Structure of a Neuron 11
2.2 Typical ANN 12
2.3 Operation of a Typical Processing Element 14
2.4 An Example of a Threshold Function 15
2.5 An Example of a Hard Limiter 16
2.6 The Sigmoid Function 17
2.7 The Hyperbolic Tangent Function 17
2.8 A Taxonomy of Neural Networks 24
2.9 Six Types of Network Architectures 25
2.10 Typical Back-Propagation Architecture 27
2.11 The Sigmoidal Transfer Function and its Derivative 29
2.12 A Typical Error Surface 31
2.13 The Phenomenon of Overtraining 35
2.14 The Phenomenon of Overfitting 38
2.15 Illustration of One Iteration Step 45
2.16 An Example of a Radial Basis Transfer Function 50
2.17 Summary of the Box-Jenkins Methodology 88
2.18 Steps in the Development of an ANN model 111

3.1 Extent of the Murray Darling Basin 119
3.2 Profile of the River Murray 120
3.3 The River Murray Barrages 121
3.4 The River Murray Pipelines 122
3.5 River Murray Salinity Profile 124
3.6 Inflows of Saline Groundwater 125
3.7 Natural Sources of Salinity 126
3.8 Salt Inflows Aggravated by River Structures 127
3.9 The Effect of Salinity on Crop Yield 130
3.10 Locations of Available Data 137
3.11 Salinity at Murray Bridge (SMB) (1987 to 1992) 137

3.12 Salinity at Murray Bridge (SMB) and Salinity at Mannum (SMN) - 1987 139
3.13 Salinity at Murray Bridge (SMB) and Salinity at Morgan (SMO) - 1987..... 140
3.14 Salinity at Murray Bridge (SMB) and Salinity at Waikerie (SWE) - 1987 140
3.15 Salinity at Murray Bridge (SMB) and Salinity at Loxton (SLO) - 1987..... 141
3.16 Salinity at Murray Bridge (SMB) and Salinity at Lock 5 Upper (SL5U) - 1987 141
3.17 Salinity at Murray Bridge (SMB) and Salinity at Mannum (SMN) - 1988 142
3.18 Salinity at Murray Bridge (SMB) and Salinity at Morgan (SMO) - 1988..... 142
3.19 Salinity at Murray Bridge (SMB) and Salinity at Waikerie (SWE) - 1988 143
3.20 Salinity at Murray Bridge (SMB) and Salinity at Loxton (SLO) - 1988..... 143
3.21 Salinity at Murray Bridge (SMB) and Salinity at Lock 5 Upper (SL5U) - 1988 144
3.22 Salinity at Murray Bridge (SMB) and Salinity at Mannum (SMN) - 1989 144
3.23 Salinity at Murray Bridge (SMB) and Salinity at Morgan (SMO) - 1989..... 145
3.24 Salinity at Murray Bridge (SMB) and Salinity at Waikerie (SWE) - 1989 145
3.25 Salinity at Murray Bridge (SMB) and Salinity at Loxton (SLO) - 1989..... 146
3.26 Salinity at Murray Bridge (SMB) and Salinity at Lock 5 Upper (SL5U) - 1989 146
3.27 Salinity at Murray Bridge (SMB) and Salinity at Mannum (SMN) - 1990 147
3.28 Salinity at Murray Bridge (SMB) and Salinity at Morgan (SMO) - 1990..... 147
3.29 Salinity at Murray Bridge (SMB) and Salinity at Waikerie (SWE) - 1990 148
3.30 Salinity at Murray Bridge (SMB) and Salinity at Loxton (SLO) - 1990..... 148
3.31 Salinity at Murray Bridge (SMB) and Salinity at Lock 5 Upper (SL5U) - 1990 149
3.32 Salinity at Murray Bridge (SMB) and Salinity at Mannum (SMN) - 1991 149
3.33 Salinity at Murray Bridge (SMB) and Salinity at Morgan (SMO) - 1991..... 150
3.34 Salinity at Murray Bridge (SMB) and Salinity at Waikerie (SWE) - 1991 150
3.35 Salinity at Murray Bridge (SMB) and Salinity at Loxton (SLO) - 1991..... 151
3.36 Salinity at Murray Bridge (SMB) and Salinity at Lock 5 Upper (SL5U) - 1991 151
3.37 Salinity at Murray Bridge (SMB) and Salinity at Mannum (SMN) - 1992 152
3.38 Salinity at Murray Bridge (SMB) and Salinity at Morgan (SMO) - 1992..... 152
3.39 Salinity at Murray Bridge (SMB) and Salinity at Waikerie (SWE) - 1992 153
3.40 Salinity at Murray Bridge (SMB) and Salinity at Loxton (SLO) - 1992..... 153
3.41 Salinity at Murray Bridge (SMB) and Salinity at Lock 5 Upper (SL5U) - 1992 154
3.42 Salinity at Murray Bridge (SMB) and Flow at Lock 1 Lower (FL1L) - 1987 to 1992..... 155

3.43 Salinity at Murray Bridge (SMB) and Flow at Overland Corner (FOC) - 1987 to 1992 155
3.44 Salinity at Murray Bridge (SMB) and River Level at Murray Bridge (LMB) - 1987 to 1992 157
3.45 Salinity at Murray Bridge (SMB) and River Level at Mannum (LMN) - 1987 to 1992 157
3.46 Salinity at Murray Bridge (SMB) and River Level at Lock 1 Lower (LL1L) - 1987 to 1992 158
3.47 Salinity at Murray Bridge (SMB) and River Level at Lock 1 Upper (LL1U) - 1987 to 1992 158
3.48 Salinity at Murray Bridge (SMB) and River Level at Morgan (LMO) - 1987 to 1992 159
3.49 Salinity at Murray Bridge (SMB) and River Level at Waikerie (LWE) - 1987 to 1992 159
3.50 Salinity at Murray Bridge (SMB) and River Level at Overland Corner (LOC) - 1987 to 1992 160
3.51 Salinity at Murray Bridge (SMB) and River Level at Loxton (LLO) - 1987 to 1992 160
3.52 ACF for Salinity at Murray Bridge (1987-1991) 168
3.53 PACF for Salinity at Murray Bridge (1987-1991) 168
3.54 Data Set 11 173
3.55 ACF for Data Set 11 173
3.56 PACF for Data Set 11 174
3.57 Typical Values of Connection Weights 177
3.58 Actual and Predicted Salinities Obtained Using Model SMB_U_A_04_02_91 - 1 Day Forecast 178
3.59 Relative Significance of Network Inputs for Model SMB_U_A_09_01_91 181
3.60 Actual and Predicted Salinities Obtained Using Model SMB_U_A_04_02_91 - 14 Day Forecast 182
3.61 Connection Weights for Model SMB_U_A_10_03_91 185
3.62 Relative Significance of Network Inputs for Model SMB_U_A_11_03_91 187
3.63 Actual and Predicted Salinities Obtained Using Model SMB_U_A_10_03_91 - 1 Day Forecast - No Hidden Layers 187
3.64 Actual and Predicted Salinities Obtained Using Model SMB_U_A_12_03_91 - 1 Day Forecast - Two Hidden Layers 188
3.65 Actual and Predicted Salinities Obtained Using Model SMB_U_A_10_03_91 - 14 Day Forecast (Direct Method) 190

3.66 Comparison of Connection Weights for 1 and 14 Day Forecasts for Model SMB_U_A_10_03_91 (No Hidden Layers) 191
3.67 Comparison of Relative Significance of Inputs for 1 and 14 Day Forecasts for Model SMB_U_A_11_03_91 (Two Hidden Layers) 192
3.68 Actual and Predicted Salinities Obtained Using Model SMB_U_A_12_03_91 - 14 Day Forecast (Direct Method) 193
3.69 Actual and Predicted Salinities Obtained Using Model SMB_U_A_10_03_91 - 3 Day Forecast (Recursive Method) 195
3.70 Actual and Predicted Salinities Obtained Using Model SMB_U_A_10_03_91 - 5 Day Forecast (Recursive Method) 196
3.71 Actual and Predicted Salinities Obtained Using Model SMB_U_A_10_03_91 - 14 Day Forecast (Recursive Method) 196
3.72 Actual and Predicted Salinities Obtained Using Model SMB_U_A_12_03_91 - 14 Day Forecast (Recursive Method) 198
3.73 Actual and Predicted Salinities Obtained Using Model SMB_U_A_13_04_91 - 14 Day Forecast (Recursive Method) 199
3.74 Real Time Forecast of Salinity in the River Murray at Murray Bridge Obtained Using Model SMB_U_A_15_06_91 - 1991 - 14 Days in Advance 202
3.75 Procedure for Determining Appropriate Model Inputs 204
3.76 Typical Sensitivities of Salinity Inputs from Mannum 210
3.77 Best 14 Day Forecast of Salinity at Murray Bridge for 1988 - Model SMB_M_A_18_10_88 215
3.78 Best 14 Day Forecast of Salinity at Murray Bridge for 1989 - Model SMB_M_A_18_10_89 215
3.79 Best 14 Day Forecast of Salinity at Murray Bridge for 1990 - Model SMB_M_A_18_10_90 216
3.80 Best 14 Day Forecast of Salinity at Murray Bridge for 1991 - Model SMB_M_A_18_10_91 216
3.81 Summary of the Back-Propagation Training Process 218
3.82 RMSE at Different Stages of Learning for Various Epoch Sizes 221
3.83 RMSE at Different Stages of Learning for Various Epoch Sizes (Up to a Learn Count of 400,000) 222
3.84 Change in RMSE versus Number of Weight Updates for Models Using Different Epoch Sizes 223
3.85 Change in RMSE versus Normalised Number of Weight Updates for Models SMB_M_A_22_10_88 and SMB_M_A_26_10_88 224
3.86 RMSE at Different Stages of Learning for Various Epoch Sizes ($\eta = 0.1$, $\mu = 0.6$) 225

3.87 RMSE at Different Stages of Learning for Various Epoch Sizes ($\eta = 0.005$, $\mu = 0.05$) 226
3.88 RMSE at Different Stages of Learning for Various Values of Momentum ($\eta = 0.005$) 228
3.89 RMSE at Different Stages of Learning for Various Values of Momentum ($\eta = 0.1$) 229
3.90 RMSE at Different Stages of Learning for Various Values of Momentum ($\varepsilon = 5$) 231
3.91 RMSE at Different Stages of Learning for Various Values of Momentum ($\varepsilon = 16$) 232
3.92 RMSE at Different Stages of Learning for Various Values of Momentum ($\varepsilon = 365$) 233
3.93 RMSE at Different Stages of Learning for Various Learning Rates ($\mu = 0.05$) 235
3.94 RMSE at Different Stages of Learning for Various Learning Rates ($\mu = 0.05$) (Up to a Learn Count of 200,000) 236
3.95 Example of an Error Surface in the Vicinity of Local Minima 237
3.96 RMSE at Different Stages of Learning for Various Learning Rates ($\mu = 0.4$) 238
3.97 RMSE at Different Stages of Learning for Various Learning Rates ($\mu = 0.6$) 239
3.98 RMSE at Different Stages of Learning for Various Learning Rates ($\mu = 0.8$) 240
3.99 RMSE at Different Stages of Learning for Various Learning Rates ($\mu = 0.9$) 241
3.100 RMSE at Different Stages of Learning for Various Learning Rates ($\varepsilon = 5$) 242
3.101 RMSE at Different Stages of Learning for Various Learning Rates ($\varepsilon = 10$) 243
3.102 RMSE at Different Stages of Learning for Various Learning Rates ($\varepsilon = 16$) 244
3.103 RMSE at Different Stages of Learning for Various Learning Rates ($\varepsilon = 35$) 245
3.104 RMSE at Different Stages of Learning for Models Using a Learning Rate of 0.005 and the Delta-Bar-Delta Algorithm 246
3.105 RMSE at Different Stages of Learning for Models with Different Transfer Functions 248
3.106 RMSE at Different Stages of Learning for Models with Different Error Functions 249

3.107 RMSE at Various Stages of Learning for Different Ratios of First to Second Hidden Layer Nodes 252
3.108 RMSE at Various Stages of Learning for Different Numbers of Nodes in the Hidden Layers 253
3.109 RMSE at Various Stages of Learning for Different Numbers of Nodes in the First Hidden Layer 254
3.110 RMSE at Various Stages of Learning for Model SMB_M_A_72_10_88 259
3.111 RMSE at Various Stages of Learning for Model SMB_M_A_72_10_88 (Up to a Learn Count of 25,000) 260
3.112 Actual and Predicted Salinities at a Learn Count of 1,000 - Model SMB_M_A_72_10_88 261
3.113 Actual and Predicted Salinities at a Learn Count of 3,000 - Model SMB_M_A_72_10_88 262
3.114 Actual and Predicted Salinities at a Learn Count of 6,000 - Model SMB_M_A_72_10_88 262
3.115 Actual and Predicted Salinities at a Learn Count of 9,000 - Model SMB_M_A_72_10_88 263
3.116 Actual and Predicted Salinities at a Learn Count of 320,000 - Model SMB_M_A_72_10_88 264
3.117 Actual and Predicted Salinities at a Learn Count of 3,400,000 - Model SMB_M_A_72_10_88 264
3.118 Changes in RMS Prediction Error After Each Weight Update for Model SMB_M_A_19_11_88 267
3.119 Typical Plot Showing the Phenomenon of Whole Portions of the Plot of Predicted Salinities Moving Up or Down from One Weight Update to the Next 268
3.120 RMSE at Different Stages of Learning for Models With "Spiked" Training / Testing Sets ($\varepsilon = 5$) 270
3.121 RMSE at Different Stages of Learning for Models With "Spiked" Training / Testing Sets ($\varepsilon = 365$) 271
3.122 Example of an Error Surface that Enables Local Minima to be Escaped when Small Steps are Taken in Weight Space 273
3.123 Typical Minimum "Plateau" in the Error Surface for Large Networks 276
3.124 Typical Generalisation Ability of a Neural Network Model During Training .. 276
3.125 Effect of the Size of the Steps Taken in Weight Space on Generalisation Ability and Network Behaviour 278
3.126 Best Results Obtained When Using Various Momentums at a Given Learning Rate 281

3.127 Best Results Obtained When Using Various Learning Rates at a Given Momentum 281
3.128 Best Results Obtained for Networks with One and Two Hidden Layers 282
3.129 Real Time Forecast of Salinity in the River Murray at Murray Bridge Obtained Using Model SMB_M_A_79_13_91 - 1991 - 1 Day in Advance 287
3.130 Real Time Forecast of Salinity in the River Murray at Murray Bridge Obtained Using Model SMB_M_A_79_13_91 - 1991 - 5 Days in Advance ... 288
3.131 Real Time Forecast of Salinity in the River Murray at Murray Bridge Obtained Using Model SMB_M_A_79_13_91 - 1991 - 14 Days in Advance 288
3.132 Data Set 12 292
3.133 ACF for Data Set 12 292
3.134 PACF for Data Set 12 293
3.135 Actual and Predicted Salinities - Model SMB_U_T_06_12_91 - 1 Day in Advance 304
3.136 Actual and Predicted Salinities - Model SMB_U_T_10_12_91 - 1 Day in Advance 304
3.137 Actual and Predicted Salinities - Model SMB_U_T_03_11_91 - 1 Day in Advance 305
3.138 Actual and Predicted Salinities - Model SMB_U_T_03_11_91 - 3 Days in Advance 305
3.139 Actual and Predicted Salinities - Model SMB_U_T_03_11_91 - 5 Days in Advance 306
3.140 Actual and Predicted Salinities - Model SMB_U_T_08_12_91 - 1 Day in Advance (AR Parameters of Order 1, 365 and 366 & MA Parameter of Order 1) 307
3.141 Actual and Predicted Salinities - Model SMB_U_T_09_12_91 - 1 Day in Advance (AR Parameters of Order 1 and 365 & MA Parameter of Order 1) 307
3.142 Actual and Predicted Salinities - Model SMB_U_T_04_11_91 - 3 Days in Advance (Differencing of Order D = 1) 308
3.143 Actual and Predicted Salinities - Model SMB_U_T_08_12_91 - 3 Days in Advance (Differencing of Order D = 2) 308
3.144 Actual and Predicted Salinities - Model SMB_U_T_04_11_91 - 14 Days in Advance 309
3.145 Actual and Predicted Salinities (Filtered) - Model SMB_U_T_04_11_91 - 14 Days in Advance ($\xi = 4$) 312
3.146 Steps in the Development of the VARIMA Model 314

3.147 Actual and Predicted Salinities - Combined Model C12 - 1 Day in Advance 352
3.148 Actual and Predicted Salinities - Combined Model C12 - 3 Days in Advance 352
3.149 Actual and Predicted Salinities - Combined Model C12 - 5 Days in Advance 353
3.150 Actual and Predicted Salinities - Combined Model C12 - 14 Days in Advance 355
3.151 Actual and Predicted Salinities (Filtered) - Combined Model C12 - 14 Days in Advance ($\xi = 4$) 356
3.152 Comparison of the RMS Forecasting Errors Obtained for 1991 Using the Best UANN and MANN models (Direct Method of Forecasting) as well as a Naive Model 361
3.153 Comparison of the RMS Forecasting Errors Obtained for 1991 Using the Best ARIMA, VARIMA and Naive Models 362
3.154 Effect of Filtering on the 14-Day Forecasts (1991) Obtained Using the ARIMA and VARIMA Models 362
3.155: Comparison of the RMS Forecasting Errors Obtained for 1991 Using the Best UANN (Direct Method of Forecasting), ARIMA and Naive Models 363
3.156 Effect of Filtering on the 14-Day Forecasts (1991) Obtained Using the ARIMA Model 364
3.157 Comparison of the RMS Forecasting Errors Obtained for 1991 Using the Best MANN, VARIMA and Naive Models 366
3.158 Effect of Filtering on the 14-Day Forecasts (1991) Obtained Using the VARIMA Model 366
3.159 Actual and Predicted (14 and 28 Days in Advance) Salinities in the River Murray at Murray Bridge - 1988 to 1991 372
3.160 Actual and Predicted Salinities (28 Days Ahead) Using Actual (FL7L_0) and Predicted (FL7L_14) Flows at Lock 7 Lower 375
3.161 Real Time Forecast of Salinity in the River Murray at Murray Bridge Obtained Using Model SMB_M_A_81_15_91 - 1991 - 28 Days in Advance 377
3.162 Change in the RMSE of the Real Time Forecasts of Salinity in the River Murray at Murray Bridge with Forecasting Period (1991) 378
3.163 Typical Plot of the Relative Significance of the Model Inputs at Various Lags - 1 Day Forecast 378
3.164 Typical Plot of the Relative Significance of the Model Inputs at Various Lags - 5 Day Forecast 379

3.165 Typical Plot of the Relative Significance of the Model Inputs at Various Lags - 14 Day Forecast 379
3.166 Typical Plot of the Relative Significance of the Model Inputs at Various Lags - 28 Day Forecast 380
3.167 Outline of Neural Network Based Approach (Method 2) 383
3.168 Actual and Predicted Salinities - Model SMB_U_A_82_17_88 (i.e. Model Using Salinities at Murray Bridge as Inputs) - 14 Days in Advance 386
3.169 Actual and Predicted Salinities - Model SMB_M_A_83_20_88 (i.e. Model Using Salinities at Waikerie as Inputs) - 14 Days in Advance 386
3.170 Actual and Predicted Salinities - Model SMB_M_A_84_22_88 (i.e. Model Using Flows at Lock 1 Lower as Inputs) - 14 Days in Advance 387
3.171 Relative Significance of Inputs - Model SMB_U_A_82_17_88 (i.e. Model Using Salinities at Murray Bridge as Inputs) - 14 Days in Advance 388
3.172 Relative Significance of Inputs - Model SMB_M_A_83_21_88 (i.e. Model Using Salinities at Loxton as Inputs) - 14 Days in Advance 389
3.173 Relative Significance of Inputs - Model SMB_M_A_84_22_88 (i.e. Model Using Flows at Lock 1 Lower as Inputs) - 14 Days in Advance 390
3.174 Relative Significance of Inputs - Model SMB_M_A_85_23_88 (i.e. Model Using Flows at Lock 1 Lower as Inputs) - 14 Days in Advance 391
3.175 Best Real Time Forecasts and Corresponding Training Times For Models With Inputs Obtained Using Methods 1, 2 and 3 395
3.176 Comparison of the RMSEs of the Best Real Time Forecasts Obtained for 1991 Using the ARIMA, VARIMA, UANN, MANN and NAIVE Models 401

4.1 Examples of the Four Main Groups of Algae 404
4.2 Location of Port Pirie, Port Augusta and Whyalla in Relation to Morgan 423
4.3 Concentrations of *Anabaena* at Morgan (1983 to 1993) 425
4.4 Concentrations of *Anabaena* and Colour at Morgan (1983 to 1993) 426
4.5 Concentrations of *Anabaena* and Turbidity at Morgan (1983 to 1993) 427
4.6 Concentrations of *Anabaena* and Temperature at Morgan (1983 to 1993) 427
4.7 Concentrations of *Anabaena* at Morgan and Flow at Lock 7 (1983 to 1993) 428

4.8 Concentrations of *Anabaena* and Total Phosphorus at Morgan (1983 to 1993) 429
4.9 Total Phosphorus and Turbidity at Morgan (1983 to 1993) 429
4.10 Concentrations of *Anabaena* and Soluble Phosphorus at Morgan (1983 to 1993) 430
4.11 Concentrations of *Anabaena* and Nitrogen at Morgan (1983 to 1993) 431
4.12 Two-Week Forecasts Obtained Using the Model With Colour Inputs (Model 01-01) 435
4.13 Two-Week Forecasts Obtained Using the Model With Turbidity Inputs (Model 01-02) 435
4.14 Two-Week Forecasts Obtained Using the Model With Temperature Inputs (Model 01-03) 436
4.15 Two-Week Forecasts Obtained Using the Model With Flow Inputs (Model 02-04) 436
4.16 Two-Week Forecasts Obtained Using the Model With Soluble Phosphorus Inputs (Model 01-05) 437
4.17 Two-Week Forecasts Obtained Using the Model With Total Phosphorus Inputs (Model 02-06) 437
4.18 Two-Week Forecasts Obtained Using the Model With Nitrogen Inputs (Model 01-07) 438
4.19 Relative Significance of Inputs - Model 01_01 (i.e. Model Using Colour Inputs) 442
4.20 Relative Significance of Inputs - Model 01_02 (i.e. Model Using Turbidity Inputs) 443
4.21 Relative Significance of Inputs - Model 01_03 (i.e. Model Using Temperature Inputs) 443
4.22 Relative Significance of Inputs - Model 02_04 (i.e. Model Using Flow Inputs) 444
4.23 Relative Significance of Inputs - Model 01_05 (i.e. Model Using Soluble Phosphorus Inputs) 444
4.24 Relative Significance of Inputs - Model 02_06 (i.e. Model Using Total Phosphorus Inputs) 445
4.25 Relative Significance of Inputs - Model 01_07 (i.e. Model Using Nitrogen Inputs) 445
4.26 Actual and Predicted Concentrations of *Anabaena* at Morgan (2-Week Forecast) - Model With 7 Inputs 449
4.27 Actual and Predicted Concentrations of *Anabaena* at Morgan (2-Week Forecast) - Model With 43 Inputs 450

4.28 Actual and Predicted Concentrations of *Anabaena* at Morgan (2-Week Forecast) - Model With 102 Inputs 450
4.29 Actual and Predicted Concentrations of *Anabaena* at Morgan (4-Week Forecast) - Model With 102 Inputs 451
4.30 Typical Plot of the Relative Significance of the Model Inputs for the Model with 102 Inputs 452

List of Tables

3.1 Entitlement Flows to South Australia 121
3.2 Available Data (Daily) 136
3.3 Statistics of Salinity Time Series at, and Upstream of, Murray Bridge (1987 to 1992) 138
3.4 Statistics of Flow Time Series (1987 to 1992) 154
3.5 Statistics of River Level Time Series (1987 to 1992) 156
3.6 Computer Programs Developed for Time Series Modelling and their Function (Utility) 162
3.7 Computer Programs Developed for Time Series Modelling and their Function (Univariate) 163
3.8 Computer Programs Developed for Time Series Modelling and their Function (Multivariate) 164
3.9 Degrees of Seasonal and Non-Seasonal Differencing Used 169
3.10 Lags at which Values of the ACF and PACF are Significantly Different from Zero for the Differenced Data Sets Without Seasonal Differencing (D=0) 170
3.11 Lags at which Values of the ACF and PACF are Significantly Different from Zero for the Differenced Data Sets With a Degree of Seasonal Differencing Equal to 1 (D=1) 171
3.12 Lags at which Values of the ACF and PACF are Significantly Different from Zero for the Differenced Data Sets With a Degree of Seasonal Differencing Equal to 2 (D=2) 171
3.13 Connection Weights at Different Stages of Learning - Model SMB_U_A_01_01_91 ($\eta = 0.001$) 176
3.14 Connection Weights at Different Stages of Learning - Model SMB_U_A_02_01_91 ($\eta = 0.005$) 176
3.15 Connection Weights at Different Stages of Learning - Model SMB_U_A_03_01_91 ($\eta = 0.02$) 177
3.16 Best 1 Day Forecast Obtained Using Model SMB_U_A_04_02_91 178

3.17 Best 1 Day Forecasts Obtained Using Models SMB_U_A_04_02_91 and SMB_U_A_05_02_91 179
3.18 Best 1 Day Forecasts Obtained Using Models SMB_U_A_05_02_91, SMB_U_A_06_02_91 and SMB_U_A_07_02_91 179
3.19 Best 1 Day Forecasts Obtained Using Models SMB_U_A_06_02_91 and SMB_U_A_08_02_91 180
3.20 Best 1 Day Forecasts Obtained Using Models SMB_U_A_08_02_91 and SMB_U_A_09_01_91 180
3.21 Best 14 Day Forecasts Obtained Using Models SMB_U_A_04_02_91 and SMB_U_A_06_02_91 183
3.22 Best 1 Day Forecasts Obtained Using Models SMB_U_A_04_02_91 and SMB_U_A_10_03_91 184
3.23 Best 1 Day Forecasts Obtained Using Models SMB_U_A_10_03_91, SMB_U_A_11_03_91 and SMB_U_A_12_03_91 186
3.24 Best 1 Day Forecasts Obtained Using Models SMB_U_A_12_03_91 and SMB_U_A_13_04_91 188
3.25 Best 14 Day Forecasts Obtained Using Models SMB_U_A_10_03_91 (direct) and SMB_U_A_04_02_91 (recursive) 189
3.26 Best 14 Day Forecasts Obtained Using Models SMB_U_A_10_03_91, SMB_U_A_11_03_91 and SMB_U_A_12_03_91 (Direct Method) 191
3.27 Best 14 Day Forecasts Obtained Using Model SMB_U_A_12_03_91 (Direct Method) and a Naive Model 192
3.28 Best 14 Day Forecasts Obtained Using Models SMB_U_A_11_03_91 and SMB_U_A_14_05_91 (Direct Method) 193
3.29 Best 14 Day Forecasts Obtained Using Models SMB_U_A_12_03_91 and SMB_U_A_13_04_91 (Direct Method) 194
3.30 Comparison of the Best 14 Day Forecasts Obtained Using Models SMB_U_A_04_02_91 and SMB_U_A_10_03_91 (Recursive Method) 195
3.31 Best 14 Day Forecasts Obtained Using Models SMB_U_A_10_03_91 and SMB_U_A_12_03_91 (Recursive Method) 197
3.32 Best 14 Day Forecasts Obtained Using Models SMB_U_A_10_03_91 and SMB_U_A_13_04_91 (Recursive Method) 198
3.33 Best One Day Forecasts Obtained Using the Various Forecasting Methods 200
3.34 Best 14 Day Forecasts Obtained Using the Various Forecasting Methods 200
3.35 Real Time Forecasts Obtained Using Model SMB_U_A_15_06_91 202
3.36 Approximate Salinity Travel Times 205
3.37 Lags of Salinity Inputs Chosen to Cater for Salinity Transport 206
3.38 Training / Testing Sets Used 207

3.40 Best 14 Day Forecasts Obtained for the Various Training / Testing Sets 209
3.41 Results Obtained Using a Different Number of Outputs 214
3.42 Best Results for Models With Various Epoch Sizes ($\eta = 0.005$, $\mu = 0.6$) 220
3.43 Best Results for Models With Various Epoch Sizes ($\eta = 0.1$, $\mu = 0.6$) 225
3.44 Best Results for Models With Various Epoch Sizes (η=0.1, μ=0.05) 226
3.45 Summary of Models Developed For Assessing the Effects of Different Learning Rates and Momentum Values 227
3.46 Best Results for Models With Various Values of Momentum ($\eta = 0.005$, $\varepsilon = 16$) 228
3.47 Best Results for Models With Various Values of Momentum ($\eta = 0.02$, $\varepsilon = 16$) 230
3.48 Best Results for Models With Various Values of Momentum ($\eta = 0.1$, $\varepsilon = 16$) 230
3.49 Best Results for Models With Various Values of Momentum ($\eta = 0.2$, $\varepsilon = 16$) 230
3.50 Best Results for Models With Various Values of Momentum ($\varepsilon = 5$, $\eta = 0.005$) 232
3.51 Best Results for Models With Various Values of Momentum ($\varepsilon = 16$, $\eta = 0.005$) 232
3.52 Best Results for Models With Various Values of Momentum ($\varepsilon = 365$, $\eta = 0.005$) 233
3.53 Best Results for Models With Various Learning Rates ($\mu = 0.05$, $\varepsilon = 16$) 236
3.54 Best Results When Using Various Learning Rates ($\mu = 0.4$, $\varepsilon = 16$) 238
3.55 Best Results When Using Various Learning Rates ($\mu = 0.6$, $\varepsilon = 16$) 239
3.56 Best Results When Using Various Learning Rates ($\mu = 0.8$, $\varepsilon = 16$) 240
3.57 Best Results When Using Various Learning Rates ($\mu = 0.9$, $\varepsilon = 16$) 241
3.58 Best Results When Using Various Learning Rates ($\varepsilon = 5$, $\mu = 0.6$) 242
3.59 Best Results When Using Various Learning Rates ($\varepsilon = 10$, $\mu = 0.6$) 243
3.60 Best Results When Using Various Learning Rates ($\varepsilon = 16$, $\mu = 0.6$) 244
3.61 Best Results When Using Various Learning Rates ($\varepsilon = 35$, $\mu = 0.6$) 245
3.62 Comparison of the Results Obtained Using a Learning Rate of 0.005 and the Delta-Bar-Delta Algorithm (dbd) 246
3.63 Best Results Using Various Transfer Functions 248
3.64 Best Results Using Various Error Functions 249
3.65 Best Results Using Different Initial Weight Distributions 250
3.66 Best Results Using Different Ratios of First to Second Hidden Layer Nodes 252
3.67 Best Results Using Different Numbers of Hidden Layer Nodes 253

3.68 Best Results Using Different Numbers of Nodes in the First Hidden Layer 255
3.69 Optimum Network Parameters 255
3.70 Comparison of the Best Results Obtained for Models with One and Two Hidden Layers and Learning Rates of 0.005 and 0.1 256
3.71 Comparison of Best Results Obtained for Model SMB_M_A_71_10_88 when Using Different Testing Frequencies 256
3.72 Best Results Obtained Using Default and "Optimum" Settings 257
3.73 Best Result Obtained Using Model SMB_M_A_72_10_88 258
3.74 Best Results Obtained When Using "Spiked" Training / Testing Sets ($\varepsilon = 5$) 270
3.75 Best Results Obtained When Using "Spiked" Training / Testing Sets ($\varepsilon = 365$) 271
3.76 Result Obtained When Applying the Procedure for Reaching a Local Minimum in the Error Surface to model SMB_M_A_18_10_88 274
3.77 RMS Prediction Errors (EC units) After Jogging 275
3.78 Real Time Forecasts Obtained Using Model SMB_M_A_76_12_91 (Default Values of Learning Rate and Momentum) 285
3.79 Real Time Forecasts Obtained Using Model SMB_M_A_77_12_91 ($\eta = 0.02$, $\mu = 0.6$) 286
3.80 Real Time Forecasts Obtained Using Model SMB_M_A_78_12_91 (One Hidden Layer of Nodes) 286
3.81 Real Time Forecasts Obtained Using Model SMB_M_A_79_13_91 (No Inputs of Future Flows and Levels) 287
3.82 Number and Order of MA and AR Parameters for the Models using Data Set 11 294
3.83 Number and Order of MA and AR Parameters for the Models using Data Set 12 295
3.84 Final Estimates of Model Parameters for Model SMB_U_T_01_11_91 296
3.85 Final Estimates of Model Parameters for Model SMB_U_T_02_11_91 296
3.86 Final Estimates of Model Parameters for Model SMB_U_T_03_11_91 296
3.87 Final Estimates of Model Parameters for Model SMB_U_T_04_11_91 296
3.88 Final Estimates of Model Parameters for Model SMB_U_T_05_12_91 297
3.89 Final Estimates of Model Parameters for Model SMB_U_T_06_12_91 298
3.90 Final Estimates of Model Parameters for Model SMB_U_T_07_12_91 298
3.91 Final Estimates of Model Parameters for Model SMB_U_T_08_12_91 299
3.92 Final Estimates of Model Parameters for Model SMB_U_T_09_12_91 299
3.93 Final Estimates of Model Parameters for Model SMB_U_T_10_12_91 299
3.94 Values of Q^{BP} and $\chi^2_{(5)}$ for the Models using Data Set 11 300

3.95 Values of Q^{BP} and $\chi^2_{(5)}$ for the Models using Data Set 12 300
3.96 Results for Data Set 11 302
3.97 Results for Data Set 12 303
3.98 Effect of Less-Significant Parameters on Model Performance 306
3.99 Filter Weights for Various Values of ξ (Symmetric Filter) 310
3.100 Filtered 14 Day Forecasts for Model SMB_U_T_04_11_91 (Symmetrical Filter) 310
3.101 Filter Weights for Various Values of ξ (One-sided Filter) 311
3.102 Filtered 14 Day Forecasts for Model SMB_U_T_04_11_91 (One-sided Filter) 311
3.103 Behaviour of the ACF and PACF of the Component Time Series 315
3.104 Lags at which Values of the ACF and PACF are Significantly Different from Zero for the Differenced Salinity Time Series 316
3.105 Lags at which Values of the ACF and PACF are Significantly Different from Zero for the Differenced Level Time Series 317
3.106 Lags at which Values of the ACF and PACF are Significantly Different from Zero for the Differenced Flow Time Series 318
3.107 Initial Choice of Parameter Types and Corresponding Lags for each ARIMA Model 319
3.108 Salinity at Mannum - Model Parameters (Model SMN_U_T_01_11_91) 320
3.109 Salinity at Morgan - Model Parameters (Model SMO_U_T_01_11_91) 320
3.110 Salinity at Waikerie - Model Parameters (Model SWE_U_T_01_11_91) 321
3.111 Salinity at Loxton - Model Parameters (Model SLO_U_T_01_11_91) 321
3.112 Salinity at Lock 5 Upper - Model Parameters (Model SL5U_U_T_01_11_91) 321
3.113 Flow at Lock 1 Lower - Model Parameters (Model FL1L_U_T_01_11_91) 322
3.114 Level at Murray Bridge - Model Parameters (Model LMB_U_T_01_11_91) 322
3.115 Level at Mannum - Model Parameters (Model LMN_U_T_01_11_91) 322
3.116 Level at Lock 1 Lower - Model Parameters (Model LL1L_U_T_01_11_91) 322
3.117 Level at Lock 1 Upper - Model Parameters (Model LL1U_U_T_01_11_91) 323
3.118 Level at Morgan - Model Parameters (Model LMO_U_T_01_11_91) 323
3.119 Level at Waikerie - Model Parameters (Model LWE_U_T_01_11_91) 323
3.120 Level at Overland Corner - Model Parameters (Model LOC_U_T_01_11_91) 324
3.121 Level at Loxton - Model Parameters (Model LLO_U_T_01_11_91) 324

3.122 Values of Q^{BP} and $\chi^2_{(5)}$ for the Univariate Models Considered.....325
3.123 Significant Cross-Correlations Between the Residuals of Salinity at Murray Bridge and the Residuals of the Other Component Time Series.....326
3.124 Maximum Cross-Correlations Between Residuals of Level and Flow Data.....328
3.125 Order of MA and AR Parameters for the Initial Multivariate Model of Salinity at Murray Bridge (Model SMB_M_T_11_11_91).....329
3.126 Significant Cross-Correlations Between the Residuals of Salinity at Mannum and the Residuals of the Time Series Upstream of Mannum.....330
3.127 Significant Cross-Correlations Between the Residuals of Salinity at Morgan and the Residuals of the Time Series Upstream of Morgan.....330
3.128 Significant Cross-Correlations Between the Residuals of Salinity at Waikerie and the Residuals of the Time Series Upstream of Waikerie.....330
3.129 Significant Cross-Correlations Between the Residuals of Salinity at Loxton and the Residuals of the Time Series Upstream of Loxton.....330
3.130 Significant Cross-Correlations Between the Residuals of Level at Lock 1 Upper and the Residuals of the Time Series Upstream of Lock 1.....331
3.131 Order of MA and AR Parameters for the Initial Multivariate Model Predicting Salinity at Mannum (Model SMN_M_T_02_11_91).....331
3.132 Order of MA and AR Parameters for the Initial Multivariate Model Predicting Salinity at Morgan (Model SMO_M_T_02_11_91).....331
3.133 Order of MA and AR Parameters for the Initial Multivariate Model Predicting Salinity at Waikerie (Model SWE_M_T_02_11_91).....332
3.134 Order of MA and AR Parameters for the Initial Multivariate Model Predicting Salinity at Loxton (Model SLO_M_T_02_11_91).....332
3.135 Order of MA and AR Parameters for the Initial Multivariate Model Predicting Salinity at Lock 5 Upper (Model SL5U_M_T_02_11_91).....332
3.136 Order of MA and AR Parameters for the Initial Multivariate Model Predicting Level at Lock 1 Upper (Model LL1U_M_T_02_11_91).....332
3.137 Order of MA and AR Parameters for the Initial Multivariate Model Predicting Level at Loxton (Model LLO_U_T_02_11_91).....333
3.138 Estimates of Model Parameters for Models Predicting Salinity at Murray Bridge (Including MA Parameters).....340
3.139 Estimates of Model Parameters for Models Predicting Salinity at Murray Bridge (No MA Parameters).....342
3.140 Estimates of Model Parameters for Models Predicting Salinity at Mannum.....343
3.141 Estimates of Model Parameters for Models Predicting Salinity at Morgan.....344

3.142 Estimates of Model Parameters for Models Predicting Salinity at Waikerie 345
3.143 Estimates of Model Parameters for Models Predicting Salinity at Loxton..... 346
3.144 Estimates of Model Parameters for Models Predicting Salinity at Lock 5 Upper..... 347
3.145 Estimates of Model Parameters for Models Predicting Level at Loxton 347
3.146 Estimates of Model Parameters for Models Predicting Level at Lock 1 Upper..... 348
3.147 Combinations of Models Used to Obtain Multi-Step Forecasts of Salinity at Murray Bridge..... 349
3.148 Results for Combined Models C01 to C04 (Models With MA Parameters for Predicting Salinity at Murray Bridge)..... 350
3.149 Results for Combined Models C05 to C12 (Models Without MA Parameters for Predicting Salinity at Murray Bridge)..... 351
3.150 Change in Prediction Error with the Number of Parameters..... 353
3.151 Change in Prediction Error with the Omission of Dominant Parameters..... 354
3.152 Performance of Models with Different AIC Values..... 354
3.153 Filtered 14 Day Forecasts for Combined Model C12 (One-sided Filter)..... 356
3.154 Values of Q^{BP} and $\chi^2_{(5)}$ for the Multivariate Models Used for Forecasting..... 358
3.155 Values of Q^{BP} and $\chi^2_{(5)}$ for the Cross-Correlations Between the Residuals For Combined Model C12..... 358
3.156 Q^{BP}-matrix for Combined Model C12..... 359
3.157 Comparison of the Coefficients Obtained for Model SMB_U_T_04_11_91 with the Weights Obtained for Model SMB_U_A_04_02_91..... 364
3.158 Comparison of 1 and 14 Day Forecasts Obtained Using Models SMB_U_T_04_11_91 and SMB_U_A_04_02_91..... 365
3.159 Inputs Used for Forecasting Salinity at Murray Bridge 28 Days in Advance (Training / Testing Set 14)..... 369
3.160 Best 14 and 28 Day Forecasts Obtained Using Various Training / Testing Sets..... 371
3.161 Real Time Forecasts Obtained 28 Days in Advance..... 376
3.162 Model Inputs Obtained Using the Method of Haugh and Box (Method 1)..... 382
3.163 Details of the Univariate and Bivariate Models Trained for Input Identification Purposes..... 384
3.164 Best Results Obtained Using the Various Univariate and Bivariate Models - 14 Day Forecasts..... 385

3.165 Model Inputs Obtained Using the Method Based on a Neural Network Approach (Method 2) 389
3.166 Comparison of Inputs Obtained Using the Various Methods 392
3.167: Comparison of the Number of Inputs Obtained Using the Various Methods 393
3.168 Real Time Forecasting Errors for Models Using Inputs Obtained by Methods 1 (Training / Testing Set 24), 2 (Training / Testing Set 25) and 3 (Training / Testing Set 26) 394

4.1 Summary of Various Cyanobacterial Toxins and Effects 406
4.2 Odour Causing Compounds and their Effects 411
4.3 Effectiveness of Various Oxidants in Toxin Removal 414
4.5 Statistics of the Available Time Series (1983 to 1993) 424
4.6 Details of the Bivariate Models Developed for Input Identification Purposes 434
4.7 Relative Magnitude of Actual and Predicted Peak Concentrations of *Anabaena* 439
4.8 Amount by Which Each of the Inputs was Increased During the Sensitivity Analyses 441
4.9 Multivariate Model Inputs Chosen 446
4.10 Multivariate Models Trained for Each of the 3 Training / Testing Sets 447
4.11 Relative Magnitude of 2-Week Forecasts 448
4.12 Timing of 2-Week Forecasts 449

Abstract

In this research, the suitability of back-propagation artificial neural networks (ANNs) for modelling multivariate water quality time series is demonstrated. They are successfully applied to two case studies, namely the long-term forecasting of salinity, and the modelling of blue-green algae, in the River Murray, Australia.

Univariate and multivariate ANN and autoregressive integrated moving average (ARIMA) models are developed to obtain real time forecasts of salinity in the River Murray at Murray Bridge, South Australia, 1, 5 and 14 days in advance. A number of issues in relation to the construction of multivariate ANN models are investigated and guidelines for optimising their performance are developed.

The one day forecasts obtained using the ARIMA models are slightly better than those obtained using the ANN models. However, the ANN models perform significantly better than the ARIMA models with increasing forecasting period. The multivariate ARIMA model only performs marginally better than its univariate counterpart. However, multivariate ANN models perform significantly better than univariate ANN models.

Analytical methods, including the method of Haugh and Box (1977), which is commonly used to determine the inputs for multivariate time series models of the ARMA type, and a neural network based method, are proposed as means for determining the inputs for multivariate ANN models. Both methods are applied successfully to the salinity forecasting problem.

A multivariate ANN model is developed to forecast the incidence of blue-green algae (*Anabaena spp.*) in the River Murray at Morgan, South Australia, given various prevailing environmental conditions. These include the colour, turbidity, temperature and flow of the river and the concentrations of total phosphorus, soluble phosphorus and total nitrogen. The neural network based method is used to determine the lags of the model inputs. The model is able to successfully predict the relative magnitude and the timing of incidence of *Anabaena* up to 4 weeks ahead.

The interaction of the input variables is found to be very complex. No one variable is found to be dominant, with all variables contributing approximately equally. In addition, different environmental conditions are found to be responsible for significant incidences of *Anabaena*.

Statement of Originality

This work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference is made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

.DATE: 01-11-95

Acknowledgments

The author wishes to acknowledge his supervisor, Associate Professor Graeme Dandy, from the University of Adelaide, for his guidance and support throughout the duration of the project.

This project was funded by an Australian Postgraduate Research Award (APRA) and the Frank Perry Scholarship in Engineering at the University of Adelaide.

Thanks must be given to Vince Kotwicki, Bob Newman, David Bain and Phil Pfeiffer from the Engineering and Water Supply Department of South Australia (EWS) for helpful discussions and providing the data used for the salinity case study. The author would also like to thank Peter Stace from the Berri office of the EWS and Phil Suter, Peter Christie and Mike Burch from the Australian Centre for Water Quality Research for their helpful input and providing the data required for the blue-green algae case study.

The author would also like to thank Trevor Daniell, Bob Bogner and Salim Bouzerdoum from the University of Adelaide for the useful discussions on neural networks; Mark Jaksa from the University of Adelaide for his friendship and his comments on time series analysis; Ari Verbyla from the University of Adelaide for his advice on time series analysis; as well as the computing officers (Michael Brown and Byron Riessen) and office staff (Bernie, Anne, Tina and Kathy) from the Department of Civil and Environmental Engineering for their assistance.

Special thanks must also be given to my fellow postgraduate students for their support and company.

Finally, the author wishes to thank his wife Krista. Without her never ending love, patience, support, inspiration, strength, sense of humour and calming influence, the completion of this project would have been infinitely more difficult.

Chapter 1

Introduction

1.1 Introduction

Artificial neural networks (ANNs) are a form of computing inspired by the functioning of the human brain and nervous system. Although the first artificial neuron was developed in 1943 (McCulloch and Pitts, 1943), the true capabilities of ANNs have only become apparent since backpropagation networks were developed in 1986 (Rumelhart et al., 1986). The number of uses for ANNs is increasing rapidly. The world market for the commercial use of ANNs has expanded from virtual non-existence in 1985 to be in the order of tens of millions of dollars at present; and is expected to reach approximately $5 billion by the year 2000 (Hubick, 1992).

Traditionally, ANNs have been used to carry out cognitive tasks performed naturally by the brain, including recognising a familiar face, learning to speak and understand a natural language, identifying handwritten characters and determining that a target seen from different angles is in fact the same object. However, in recent years ANNs have been used extensively for forecasting and have proved to be an attractive alternative to more conventional forecasting techniques.

ANNs are particularly suited to modelling environmental systems, as they are able to perform well when a large amount of data is available, when the input data are noisy, when the probability distribution of the data is unknown, when a large number of attributes describe the system being modelled, when uncertainty exists about the rules governing the relationship between the variables and when non-linearities are suspected. In this research, ANNs have been applied to the modelling of water quality parameters in the River Murray.

The Murray-Darling system, located in the south-east of Australia, is the fourth longest river system in the world. Flow is regulated by four major storages, a series of 14 locks and weirs and 5 barrages. The Murray-Darling basin covers 1.06 million km^2, which is one seventh of Australia's surface area. Water from the River Murray is vital for irrigation as well as water supply purposes. Most major cities in South Australia are connected to the River Murray via pipelines and rely on Murray water to meet their annual water demand.

A decline in the quality of River Murray water as a result of land clearance, flow regulation, inefficient irrigation and inappropriate catchment management practices is of major concern to water management authorities. High levels of salinity and frequent outbreaks of blooms of toxic blue-green algae have a number of adverse effects on domestic, industrial, agricultural and recreational water users. Computer modelling can be used to gain a better understanding of the factors responsible for the decline in water quality and to obtain forecasts of water quality parameters to assist with the management of the Murray.

1.2 Objectives and Scope

The objectives of this research include:

1. To explore the use of ANN models for forecasting water quality parameters and to compare their performance with traditional (statistical) time series models.

2. To develop guidelines for the best construction of ANN models, addressing issues such as:
   - Possible methods for determining the inputs to multivariate ANN models.
   - The effects of less significant inputs on model performance.
   - The effects of using stationary and non-stationary data on model performance.
   - The best way to obtain multi-step forecasts.

- The behaviour of ANNs during training, especially as they approach a local minimum in the error surface.
- The effect of internal network parameters and network geometry on model performance.

The two case studies considered to achieve the above objectives include:

1. The forecasting of salinity in the River Murray at Murray Bridge, South Australia.
2. The prediction of a particular genus of blue-green algae (*Anabaena spp.*) in the River Murray at Morgan, South Australia.

The secondary objectives of the two case studies include:

i. To obtain forecasts of salinity in the River Murray at Murray Bridge up to 28 days in advance.
ii. To forecast concentrations of *Anabaena* in the River Murray at Morgan up to 4 weeks in advance.
iii. To gain a better understanding of the causal factors of blooms of blue-green algae.

1.3 Layout and Contents of Thesis

In Chapter 2, a detailed review of the modelling techniques used is given. ANNs are discussed in Section 2.1, while traditional time series analysis methods are covered in Section 2.2. In Section 2.3, the use of ANNs for modelling time series is described, and a comparison between ANN and the more traditional ARMA (Autoregressive Moving Average) type models is given.

Salinity forecasting in the River Murray is discussed in Chapter 3. An introduction to the chapter is given in Section 3.1, while a review of background literature is provided in Section 3.2. The various forecasting models used are discussed in Section 3.3.

Sections 3.4 to 3.7 give details of the various models developed for forecasting salinity in the River Murray at Murray Bridge up to 14 days in advance. The development of the univariate ANN model is covered in Section 3.4, the development of the multivariate ANN model is covered in Section 3.5, the development of the univariate time series model is covered in Section 3.6 and the development of the multivariate time series model is covered in Section 3.7. A comparison of the results obtained using the various forecasting methods is given in Section 3.8.

Long term salinity forecasting using multivariate ANN methods is explored in Section 3.9 by extending the forecasting period to 28 days.

In Section 3.10, a method for determining the inputs for multivariate ANN models is introduced and compared with empirical and analytical methods traditionally used for multivariate time series analysis. All three methods were applied to the salinity forecasting problem. The conclusions of the chapter are given in Section 3.11.

The case study of forecasting concentrations of blue-green algae in the River Murray at Morgan is discussed in Chapter 4. An introduction to the chapter is given in Section 4.1 and a review of background literature is presented in Section 4.2. The case study considered is described in Section 4.3, while the development of the multivariate ANN model is discussed in Section 4.4. The conclusions of the chapter are given in Section 4.5.

The conclusions and recommendations of the thesis are given in Chapter 5. In Section 5.1, the contributions of the research are detailed. The general and specific conclusions are provided in Sections 5.2 and 5.3 respectively. Recommendations for future work are given in Section 5.4.

Chapter 2

Literature Review

2.1 Artificial Neural Networks

2.1.1 Introduction

Artificial neural networks provide a means of computation inspired by the functioning of the human brain and nervous system.

The brain has a novel way of processing information. It utilises vast numbers of neurons whose individual processing capabilities are quite limited. These neurons are linked by, and transmit information via, electrochemical pathways. The resulting neural network is able to learn by adjusting the strength of the pathways. As a result, parallel computation can occur. The memory of the network is represented by the strength of the connections between the neurons.

Artificial neural networks (ANNs) are modelled on the gross structure of the brain outlined above. Neurons are represented by simple processing elements, which are generally arranged in layers; an input layer, an output layer and one or more layers in between called hidden layers. The processing elements are fully or partially linked by

weighted connections. Different values of weights represent connections of varying strength. These weights are adjusted using a learning rule.

Artificial neural networks are able to determine relationships between input data and corresponding output data. When presented with input-output paired training examples, artificial neural networks adjust their connection weights on the basis of the training samples and discover the rules governing the association between variables.

2.1.2 Artificial Neural Networks in Perspective

2.1.2.1 Parallel Versus Serial Computing

ANNs are a type of parallel computer, which differ from the more traditional serial (Von Neumann) computers.

Von Neumann computers are the type used most widely today. Their operation is controlled by a single central processing unit (CPU), which holds the computer's memory and processes information in a sequential manner. The sequential manner in which data is processed limits the overall processing speed of the computer, which can lead to a "bottle neck" of information transfer if a large amount of information has to be processed (Hubick, 1992).

Parallel computers consist of a number of smaller processing units that are linked together. This allows information to be processed in a parallel manner, which is useful when many pieces of information are required to carry out a particular operation (e.g. when the brain makes a decision).

2.1.2.2 ANNs as a Form of Artificial Intelligence

Artificial intelligence was first introduced in the 1950s and includes expert systems, natural language, ANNs, voice recognition, machine translation, intelligent text or media and other related activities (Hubick, 1992). A comparison of expert systems and artificial neural networks is given below.

Expert systems

Rule-based expert systems attempt to duplicate the behaviour of a human expert in a particular field of knowledge. The knowledge of an expert system is derived from a large database of knowledge which is created by a human expert. When using an expert system, this knowledge base is interrogated to obtain the desired answer, as though it

were an expert on the subject in question (Schildt, 1987; Hancox et al., 1990). The expert system also reveals the rules that were used to arrive at a particular conclusion (Hubick, 1992). The use of expert systems is well suited to serial computers.

The major advantage of expert systems is that they allow the user to check the logic used by the expert system to arrive at its conclusion (Hubick, 1992). However, there are a number of disadvantages associated with using expert systems (Hubick, 1992), including:

- A set of rules needs to be developed for each individual problem.
- If one aspect of a particular problem changes, the rules have to be adjusted. This may lead to a chain reaction, as some rules affect others and so on.
- They do not tolerate any errors in the rules.
- The rules need to be converted to the appropriate computer code.
- They are unable to cope with breakdowns or unexpected events - all possible outcomes need to be taken into account when the knowledge base is established.
- They are expensive to develop commercially.
- There is no methodology for developing the knowledge base.

Artificial neural networks

ANNs process information in a parallel manner. They develop their own set of rules as they learn "by example" and are therefore well suited for complex processes for which explicit rules are unknown (Hubick, 1992). The knowledge of ANNs is distributed and is stored in the interconnections between the processing elements. This is in contrast to expert systems, whose knowledge is represented by its rules.

The advantages of using artificial neural networks include (Jones and Hoskins, 1987; Lippmann, 1987; Daniell, 1991; Burke and Ignizio, 1992; Burke, 1991; NeuralWare, Inc., 1991; Kimoto and Asakawa, 1990; Josin, 1987; Saito and Nakano, 1990; Hecht-Nielsen, 1988; Maren et al., 1990; Hubick, 1992):

- High processing speeds as a result of their parallel nature.
- They have the ability to deal with bad or missing data.
- They are very robust or fault tolerant. If some processing elements, or the links between them, are damaged or disabled, the overall performance of the network is not degraded significantly. This property arises from the fact that information is distributed and not contained in one place (e.g. the CPU in Von Neumann computers).
- They generate their own rules internally by learning from examples. This gives neural networks an advantage when dealing with complex problems, for which the

underlying rules or algorithms are unknown. Even if an algorithm is known, it is often too computationally intensive to generate solutions in an acceptable time frame.

- They have the ability to produce continuously graded inputs or outputs. This can be used to represent the intensity of an input feature or the certainty of a classification.
- They are virtually unaffected by noisy input data.
- They can be inserted into existing technology with ease. They can be designed quickly and the direct hardware implementation of artificial neural networks is cost effective. As a result, artificial neural networks can be used for incremental system improvement and upgrade.
- They have good generalisation ability.

The disadvantages of ANNs include (Hubick, 1992):

- There is no set methodology for determining the optimal structure of the neural network for a particular problem.
- There are no means for determining the rules used by the network to arrive at a particular decision.
- They are not suited to problems where the relationship between the model inputs and outputs is deterministic.

2.1.3 History of Neural Computing

The history of neural computing is discussed by various authors (NeuralWare, Inc., 1991; Daniell, 1991; Vemuri, 1988; Maren et al., 1990; Hecht-Nielsen, 1990).

At the inception of the field of computing, there were two philosophically opposing views about what the purpose of computing should be. One philosophy stated that the mind, as well as computers, are symbol manipulating systems and that the aim of computing is to represent the computational capabilities of the brain. The opposing view was that the aim of computing is to model the structure of the brain itself (Vemuri, 1988). Neural computing follows the latter philosophy.

McCulloch and Pitts (1943) introduced one of the first theoretical models of artificial neurons, the M-P neuron. M-P neurons feature binary inputs and outputs assuming values 0 or 1 and an internal threshold level. Operation is such that the neuron output is one if the total input is greater than or equal to the threshold level. Individual M-P neurons can be interconnected to form artificial neural networks.

Hebb (1949) proposed that the connection strength between neurons changes as learning occurs. Hebb also postulated that the connection between two neurons gets stronger if the two neurons are excited simultaneously.

The first conference on artificial intelligence was held at Dartmouth in 1956. The fields of artificial intelligence and neural computing were launched at this conference (NeuralWare, Inc., 1991).

The first major research project in neural computing was conducted by Rosenblatt in 1958. Rosenblatt (1958, 1961) showed that if M-P neurons are linked with connections of adjustable strength, they can be "trained" to perform pattern classifications. These networks are called "perceptrons" and combine McCulloch and Pitts' work with Hebb's postulate. Perceptrons are able to make limited generalisations and are able to perform classifications even when the input is noisy. Perceptron networks consist of an input sensory layer, a hidden layer and an output layer. Initially, all the connection weights are given arbitrary values, producing an arbitrary response to a given input. The weights are adjusted during "training" to produce the desired output. Rosenblatt's work on perceptrons sparked great interest in the research area of neural computing and ANNs.

Widrow and Hoff (1960) developed the ADALINE (ADAptive LINear Element) learning algorithm. The ADALINE is a variant of the perceptron and is also based on simple neuron-like elements.

The two-layer version of the ADALINE, the MADALINE (Multiple ADALINE), was introduced in 1962 (Widrow and Smith, 1963). The ADALINE and MADALINE were used in a variety of applications, including weather prediction, character recognition, speech recognition and adaptive control.

Minsky and Papert (1969) published a critical review of perceptron networks called "perceptrons". They showed that single layer perceptrons cannot solve non-linear separable problems such as representing the Exclusive-OR function. As a result, funding for research into ANNs and neural computing decreased markedly and caused most research into this area to be abandoned. This is despite the fact that perceptrons with three layers of processing elements have the ability to implement any continuous function, as proven by Kolmogorov (1957). The only obstacle to using multilayer perceptrons was the lack of a suitable learning rule to adjust the weights in the hidden layer. In spite of this drawback, some research into neural networks continued.

Grossberg (1976) developed the Adaptive Resonance Theory (ART) network, which is a neurally inspired computing model.

Kohonen (1982) introduced the Self Organising Feature Map (SOM). This model is based on the principle of competitive learning, where elements compete to respond to an input stimulus and the winner adapts itself to respond more strongly to that stimulus. This results in an unsupervised learning procedure, as the internal organisation of the network is only dependent on the input stimulus.

Hopfield (1982, 1984) developed the Hopfield model, which sparked a resurgence into research in the field of neural computing. This model consists of interconnected processing elements which seek an energy minimum and store information in their connections.

In 1983, Anderson (1983) presented his Brain-State-in-a-Box model, which was based on his earlier Linear Associator model. It is a distributed, associative model based on Hebb's postulate.

The Boltzmann Machine was developed by Ackley et al. (1985). It is based on the Hopfield model, but incorporates stochastic updating procedures during recall to allow the network to escape from local minima in the energy surface. The Boltzmann Machine makes use of a search method called simulated annealing.

Rumelhart et al. (1986) introduced the Back-propagation network, which is the most widely used network today. The back-propagation learning rule enables the weights in the hidden layer(s) to be adjusted successfully, allowing back-propagation networks to implement the Exclusive-OR function.

The Bi-directional Associative Memory (BAM) network was designed by Kosko (1987) and is similar to Grossberg's models.

Another resurgence in the field of neural network research is taking place at present due to the development of new network topologies, new learning algorithms, new analog VLSI implementation techniques and a growing fascination with the functioning of the human brain (Lippmann, 1987).

2.1.4 Natural Neural Networks

The structure and operation of natural neural networks (NNNs) is addressed by several authors (NeuralWare, Inc., 1991; Vemuri, 1988; Hecht-Nielsen, 1988; Burke, 1991; Hertz et al., 1991).

NNNs consist of tens of billions of nerve cells called neurons, which are densely interconnected. Each neuron is connected to as many as 10,000 others. The cerebral cortex of the human brain is an example of a NNN.

Neurons essentially behave as microprocessors. Each neuron receives the combined output of many other neurons through input paths called dendrites. If this input signal is strong enough, the neuron is activated and produces an output which is transmitted through output structures called axons. This information transfer is electrochemical in nature.

The axon splits up and connects to the dendrites of many other neurons via junctions called synapses. The connection is chemical and is made possible by the release of neurotransmitters by the axon, which are received by the dendrites. The strength of the connection is modified as the brain learns and it depends on the amount of neurotransmitter released. The way neural networks process information, combined with the action of the synapse, results in the basic memory mechanism of the brain. The typical structure of a neuron is shown in Figure 2.1.

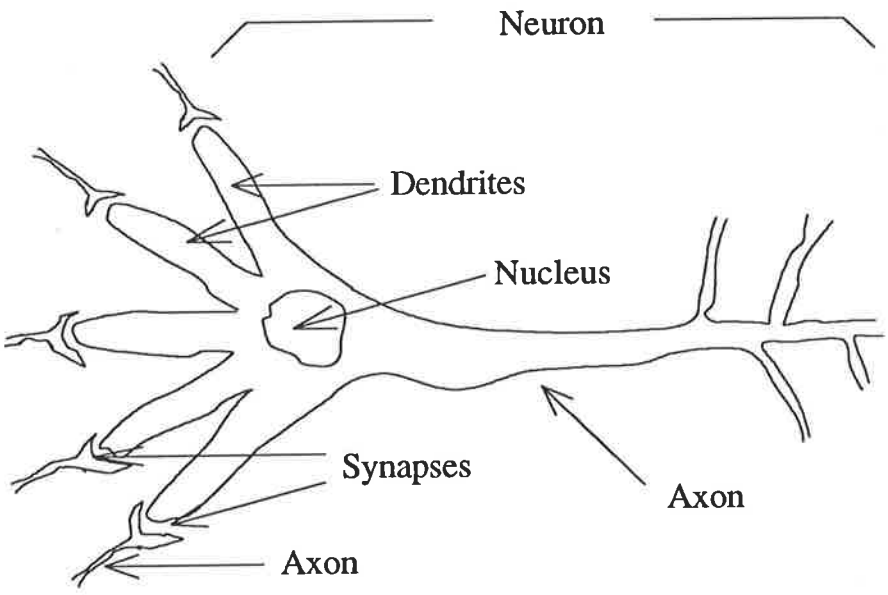


Figure 2.1: Typical Structure of a Neuron (Source: NeuralWare, Inc., 1991)

2.1.5 Artificial Neural Networks

2.1.5.1 Introduction

ANNs are described by a number of authors (NeuralWare, Inc., 1991; Lippmann, 1987; Hecht-Nielsen, 1988; Josin, 1987). They are loosely based on the structure of NNNs, but exhibit only a very small portion of their capabilities. ANNs are similar to NNNs in that they consist of interconnected processing elements (neurons) and satisfy the "locality constraint", which means that processing elements are only allowed to receive information supplied locally. As a result, the input to a processing element can only be directly affected by a node connected to its input path (Burke and Ignizio, 1992).

In ANNs, processing elements (nodes) are equivalent to neurons in natural neural networks. Processing elements are usually analog, non-linear, possess a small local memory and are slow compared with advanced digital circuitry. Individual processing elements are usually arranged in layers. Two of these layers, the input layer and the output layer, are connected to the environment. Data are presented to the network at the input layer and the response to the inputs is presented at the output layer. The layers in-between the input layer and the output layer are called hidden layers. Hidden layers enable the network to cater for non-linearities.

Nodes in the various layers are either fully or partially interconnected. Each connection has associated with it a particular adaptation coefficient or "weight", representing the synaptic strength of neural connections. A typical network is shown in Figure 2.2.

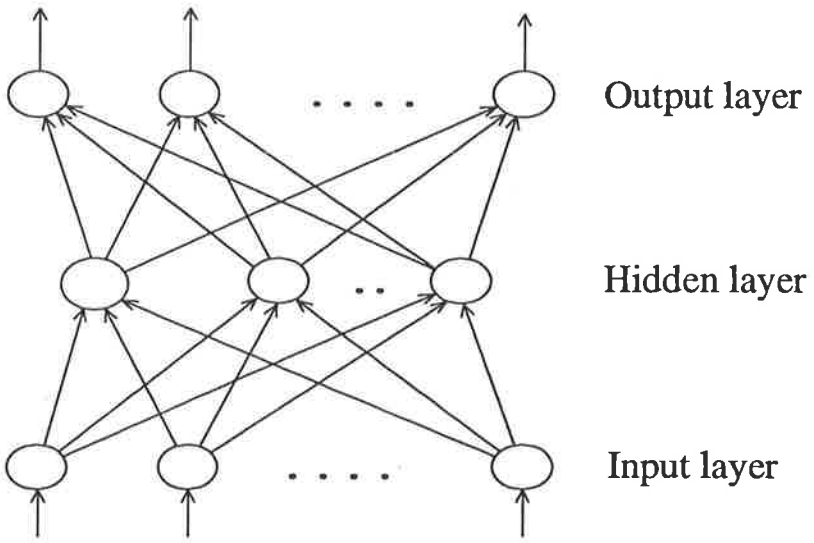


Figure 2.2: Typical ANN (Source: NeuralWare, Inc., 1991)

2.1.5.2 Operation

The operation of ANNs is described extensively by a number of authors (NeuralWare, Inc., 1991; Vemuri, 1988; Lippmann, 1987; Daniell, 1991; Maren et al., 1990; Hecht-Nielsen, 1988; Jones and Hoskins, 1987; Burke and Ignizio, 1992).

The propagation of data through the network starts with the presentation of an input stimulus at the input layer. The data then progress through, and are operated on by, the network until an output stimulus is produced at the output layer.

Individual processing elements receive inputs from many other processing elements via weighted input connections. These weighted inputs are summed and an optional threshold value is added or subtracted, producing a single activation level for the processing element. This activation level constitutes the argument of a transfer function, which produces the node output. This output is passed to the weighted input connections of many other processing elements. This process is shown for node "j" in Figure 2.3.

The following notation applies:

N^p = the number of nodes in the previous layer
x_i = the input from node i, $i = 0, 1, ..., N^p$
w_{ji} = the connection weight between nodes i and j
I_j = the activation level of node j
$F(.)$ = the transfer function
y_j = the output of node j
T'_j = the threshold for node j

The governing equations are given by (NeuralWare, Inc., 1991; Maren et al., 1990):

$$I_j = \sum_{i=1}^{N^p} w_{ji} x_i + T'_j \qquad \text{(summation)} \qquad (2.1)$$

$$y_j = F(I_j) \qquad \text{(transfer)} \qquad (2.2)$$

The performance of processing elements can be affected by changing transfer functions and adding new parameters or functions such as thresholds or gains.

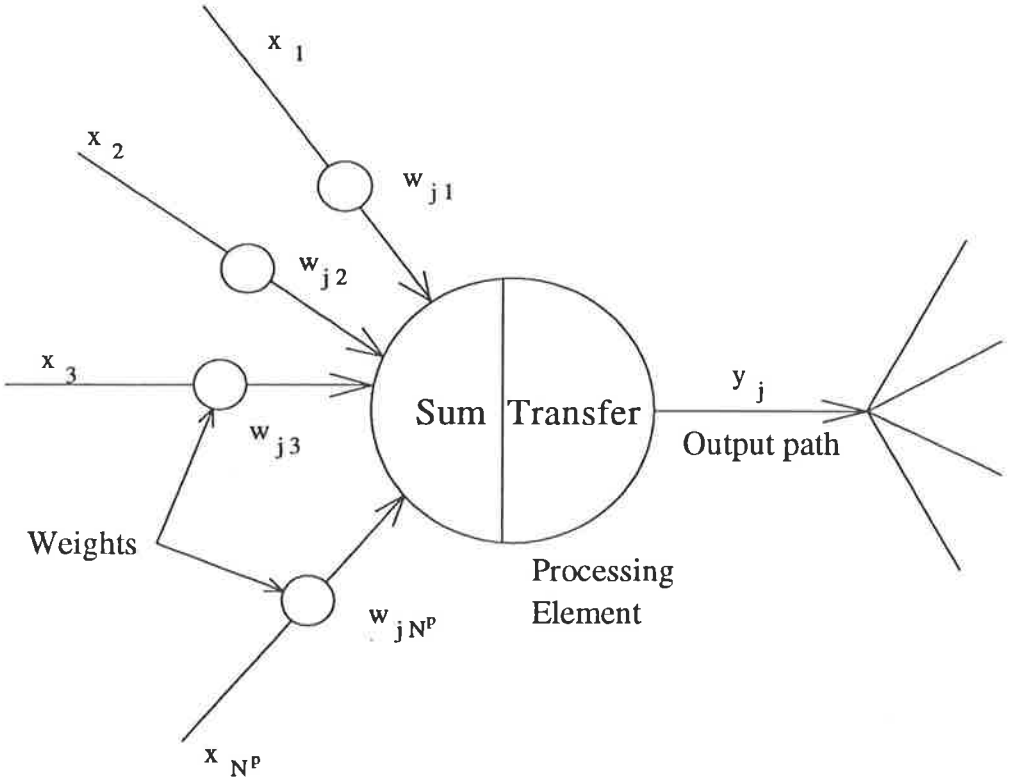


Figure 2.3: Operation of a Typical Processing Element (Source: NeuralWare, Inc., 1991)

Processing at each node occurs independently of the processing at all other nodes. At the same time, the processing done at each node effects the network as a whole as the output of one node becomes the input to many other nodes.

In a similar way to NNNs, ANNs learn by altering the connection strength between the processing elements. This is done by adjusting the weights on presentation of a set of training data using a learning rule. Once the learning phase is complete, the weights are "frozen".

Connection weights (Hecht-Nielsen, 1988; Jones and Hoskins, 1987)

Connection weights have the function of amplifying, attenuating or changing the sign of the input signal. A zero weight represents the absence of a connection and a negative weight represents an inhibitory relationship between two nodes. In general, the output of node i is multiplied by the weight of the connection between nodes i and j to produce the input signal to node j. Hence connection weights represent the strength of the connection between two nodes. Weights are stored in the local memory of nodes and hold the long-term memory of the network.

Threshold (Maren et al., 1990)

A threshold (bias) acts like another processing element that has a constant output. The effect of the threshold is to add a constant value to the summed input. The purpose of this is to scale the input to a useful range.

Transfer functions

The use of different transfer functions is addressed by various authors (Burke and Ignizio, 1992; Lippmann, 1987; Daniell, 1991; Maren et al., 1990; NeuralWare, Inc., 1991). Transfer functions are mathematical formulae that give the output of a processing element as a function of its input signal. Transfer functions can take a variety of forms.

A. Threshold functions

Threshold functions only produce an output if the total input exceeds an internal threshold (T'_j). The use of such functions is limited, as the output can only take on one of two values, as shown in Figure 2.4.

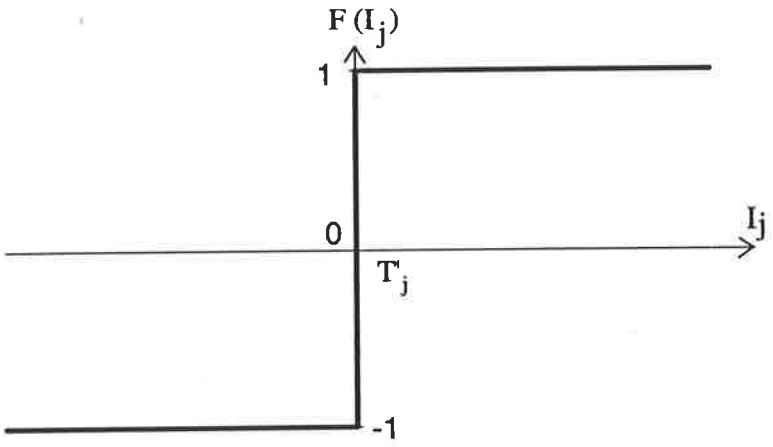


Figure 2.4: An Example of a Threshold Function (Source: Maren et al., 1990)

$$F(I_j) = \begin{cases} +1 & \text{if } I_j \geq T'_j \\ -1 & \text{otherwise} \end{cases} \qquad (2.3)$$

B. Hard limiters

Hard limiters produce a linearly varying output between defined bounds of node activation ($-\alpha$, α). Outside these bounds, the output is either -1 or 1. This is shown in Figure 2.5.

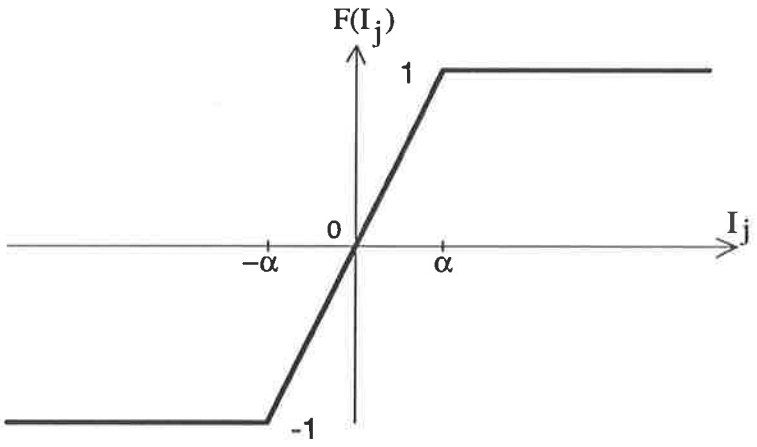


Figure 2.5: An Example of a Hard Limiter (Source: Maren et al., 1990)

$$F(I_j) = \begin{cases} +1 & \text{if} \quad I_j \geq -\alpha \\ \dfrac{I_j}{\alpha} & \text{if} \quad -\alpha < I_j < \alpha \\ -1 & \text{if} \quad I_j \leq \alpha \end{cases} \tag{2.4}$$

C. Continuous functions

Continuous functions produce a continuous output as a function of the total input. Common transfer functions of that type include the sigmoid function and the hyperbolic tangent function, both of which are examples of squashing functions. A squashing function is a special type of transfer function that is differentiable and non-decreasing. It enables a trained network to perform non-linear interpolations, to provide a reasonable output in response to a range of inputs. It also limits the output for extreme input values. The sigmoid and hyperbolic tangent transfer functions are shown in Figures 2.6 and 2.7. Sometimes other smooth functions are used as transfer functions. The sine function, for example, enables the network to perform a mode decomposition. Usually all processing elements in a particular layer have the same transfer function.

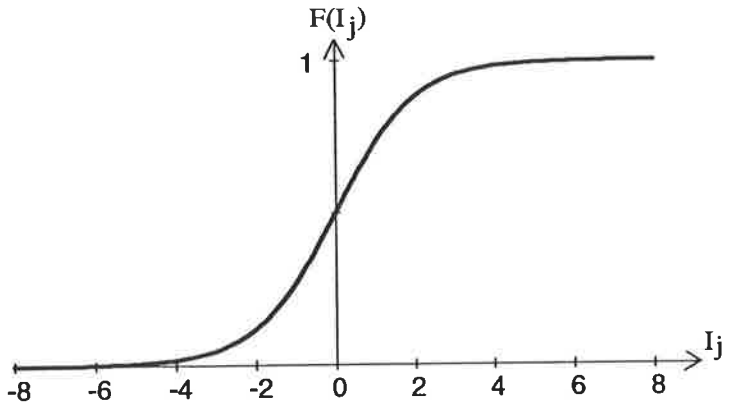


Figure 2.6: The Sigmoid Function

$$F(I_j) = \frac{1}{1+e^{-(I_j)}} \tag{2.5}$$

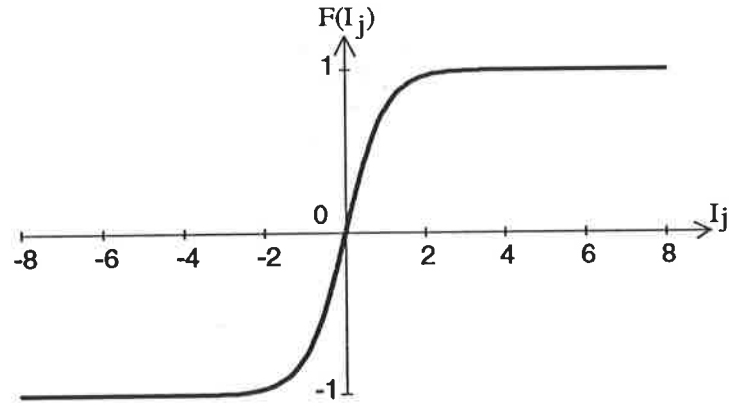


Figure 2.7: The Hyperbolic Tangent Function

$$F(I_j) = \frac{e^{(I_j)} - e^{-(I_j)}}{e^{(I_j)} + e^{-(I_j)}} \tag{2.6}$$

Learning (Hecht-Nielsen, 1988)

Learning is the process in which the weights are adjusted in response to training data provided at the input layer and, depending on the learning rule, at the output layer. The learning process allows the network to adapt its response with time in order to produce the desired output. Weights are updated in accordance with a learning rule using an on-line or off-line algorithm (Jones et al., 1990). When an off-line algorithm is used, all the training data need to be collected before learning can commence. On-

line algorithms take into account training data as they become available and are more widely used than off-line algorithms. On-line algorithms fall into two categories. Conjugate gradient learning and the multidimensional Newton's method make use of the entire training data each time the weights are updated (Jones et al., 1990). The method used by Jones et al. (1990), on the other hand, only utilises the most recent training data, resulting in increased training speed and some reduction in network accuracy.

Learning procedures can be divided into five main categories, namely unsupervised learning, supervised learning, graded learning, hybrid learning and non-adaptive learning.

A. Unsupervised learning

Unsupervised learning rules are discussed by Thorpe and Imbert (1989), Maren et al. (1990), NeuralWare, Inc. (1991) and Hecht-Nielsen (1990). In unsupervised learning, the network is only presented with input stimuli. There is no feedback from the environment to say what the corresponding output should be, or whether the output is right or wrong. The network itself discovers patterns, features, regularities or categories in the input space and reorganises itself internally so that individual hidden nodes correspond to a specific input or group of closely related inputs. Such groupings represent clusters in the input space corresponding to real life concepts. There are two different kinds of unsupervised learning, including unsupervised Hebbian learning and unsupervised competitive learning. In unsupervised Hebbian learning, several output units are often active together, whereas in unsupervised competitive learning, the processing elements compete with each other and only the winner is active. Unsupervised learning is similar to the way learning takes place in NNNs.

B. Supervised learning

Supervised learning rules are discussed by a number of authors, including Vemuri (1988), NeuralWare, Inc. (1991), Hecht-Nielsen (1988, 1990) and Maren et al. (1990). In supervised learning, the network is presented with an input stimulus as well as the desired response to that stimulus. The actual output of the network is then compared with the desired output. The aim of learning is to find a set of weights that minimises the mean squared error between the actual output and the desired output. The global error function at discrete time t, $E^G(t)$, is given by (Vemuri, 1988):

$$E^G(t) = \frac{1}{2}\sum_{k=1}^{N^O}(o_k^p(t) - o_k^d(t))^2 \qquad (2.7)$$

where

$o_k^p(t)$ = the predicted output at discrete time t

$o_k^d(t)$ = the desired (actual, historical) output at discrete time t

N^O = the number of nodes in the output layer

Initially, the weights are assigned arbitrary values. The weights are then updated systematically using a learning rule. The weight update equation typically takes the form:

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji} \qquad (2.8)$$

There are several methods for finding the weight increment, Δw_{ji}, including the following:

B1. Hebbian learning (NeuralWare, Inc., 1991; Hecht-Nielsen, 1990)

The basis for Hebbian learning is a postulate made by Hebb in 1949, which states: "When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased" (Thorpe and Imbert, 1989).

The implications for ANNs are that the weight between two processing elements is incremented if both processing elements are activated simultaneously. The weights are updated in accordance with the following equation (Vemuri, 1988):

$$\Delta w_{ji} = a_1 y_i y_j \qquad (2.9)$$

where

a_1 = a constant of proportionality

B2. Grossberg learning

Grossberg's method is another weight update procedure that is biologically plausible. The weight update equation is given as follows (Vemuri, 1988):

$$\Delta w_{ji} = a_2 f(y_j - w_{ji}) \qquad (2.10)$$

where

a_2 = a constant of proportionality

B3. Gradient descent learning (Vemuri, 1988; Maren et al., 1990)

As each combination of weights produces a different error, an error surface exists as a function of the weight space. The gradient descent method results in weights being changed in the direction of steepest descent down the error surface. As a result, the weight update is carried out in the direction that yields the maximum error reduction. The size of the step taken down the error surface is determined by a learning rate, η.

$$\Delta w_{ji} = \eta \frac{\partial E^G}{\partial w_{ji}} \qquad (2.11)$$

There is little evidence that this type of learning occurs in NNNs (Thorpe and Imbert, 1989).

B4. Competitive learning (NeuralWare, Inc., 1991)

In this type of learning, a competition occurs between all processing elements. The processing element that produces the strongest response to a particular input is the winner and is adjusted to become more like the input.

C. Graded learning

Graded learning is discussed by Maren et al. (1990), NeuralWare, Inc. (1991) and Hecht-Nielsen (1988, 1990). This type of learning is similar to supervised learning. Rather than receiving the correct response to a given stimulus, the network receives a score or grade to see how well it is performing. This reinforcement signal is only evaluative, not instructive. Graded learning is used for control and optimisation problems where it is impossible to know the correct response.

D. Hybrid learning

Sometimes it is advantageous to combine supervised and unsupervised learning techniques in the same network. The most common method of hybrid learning is to have one layer using an unsupervised learning rule followed by a network using a

supervised learning rule (Maren et al., 1990). Although the tasks performed by such a network could usually be achieved using a network that learns purely in a supervised manner, using a hybrid learning scheme speeds up the learning process considerably.

E. Non adaptive learning (Burke and Ignizio, 1992; Looi, 1992)

In this type of learning, weights are not changed adaptively, but are derived initially from an energy function. This energy function forms a representation of the problem being considered and contains information specific to that problem. While weights are fixed, node states change until the energy function reaches a steady state. The energy function must be a Liapunov function, for which lower values correspond to better solutions to the problem being represented. One such energy function, the generic Hopfield function, is given below (Burke and Ignizio, 1992):

$$E^N = -0.5 \left(\sum_i \sum_j w_{ji}\, x_i\, y_j + \sum_i T'_i\, x_i\right) \qquad (2.12)$$

2.1.5.3 Architecture

Network architecture describes the way individual processing elements are joined to form neural networks. The factors which affect network architecture include (Maren et al., 1990):

1. The number of layers (e.g. single layer, bi-layer or multi-layer).
2. The number of processing elements per layer.
3. The type of connection between individual processing elements (e.g. feed-forward connections, feedback connections, lateral connections).
4. The degree of connectivity between processing elements.

2.1.5.4 Performance

The performance of artificial neural networks is affected by three factors (Vemuri, 1988; Lippmann, 1987; Josin, 1987):

1. Node characteristics.
2. Learning rules.
3. Network architecture.

2.1.6 Implementation of Artificial Neural Networks

There are a number of ways of implementing artificial neural networks, some of which are discussed below.

2.1.6.1 Software Simulation

The operation of artificial neural networks can be simulated on conventional computers using software. Special computing languages have been developed which enable the user to describe neural network architectures. The AXON neural networks description language is one such a language (Hecht-Nielsen, 1990). A list of commercially available software simulators is given by Taylor (1993). The major disadvantage of using software simulators is that they do not make use of the inherent processing speed resulting from the parallel nature of ANNs, as processing is carried out sequentially.

2.1.6.2 Hardware Accelerators

Hardware accelerators can be used in conjunction with software simulators. The use of hardware accelerators is ideally suited to the parallel nature of neural networks. Hardware accelerators improve the processing speed of simulated networks to an extent where they can solve problems in real time. Examples of such accelerators include the Mark III and Mark IV, the Network Emulation Processor (NEP), the ANZA and ANZA Plus and the Delta-II Floating Point Processor (NeuralWare, Inc., 1991).

2.1.6.3 Optical Processors

Optical processors can also be used in the implementation of artificial neural networks and are discussed by Maren et al. (1990) and NeuralWare, Inc. (1991). Since optical processors use light, no insulation is needed, resulting in a high density of channels. In fact, the resulting density is four times as high as that obtained with electrical cables. Another advantage using optical processors is that there is no interference.

2.1.6.4 Direct Hardware Implementations

Direct hardware implementations are discussed by various authors (NeuralWare, Inc., 1991; Daniell, 1991; Hecht-Nielsen, 1988 and 1990; Maren et al., 1990).

Direct hardware implementation of artificial neural networks is desirable, as it makes full use of the parallelism of neural networks. The resulting advantages include improved processing speed and increased maximum network size. Direct hardware implementation can be achieved with the use of neurocomputers.

Neurocomputers allow parallel processing and hence implement neural networks efficiently and cost-effectively. Neurocomputers take the form of co-processor boards which plug into standard serial computers to enhance their parallel processing capabilities. Neurocomputers feature silicon chips that contain neural network architectures. Neurocomputers are of the analog, digital or hybrid type. The performance of neurocomputers can be assessed using criteria such as capacity, speed, cost, accuracy, generality, software interface provisions and configuration provisions. A list of available neurocomputers is given by Taylor (1993).

2.1.7 Different Neural Network Architectures

2.1.7.1 Network Classification

Network classification is a topic discussed by many authors (Vemuri, 1988; Lippmann, 1987; Josin, 1987; Daniell, 1991; Hecht-Nielsen, 1988; Maren et al., 1990). Many criteria can be used to classify networks of various architectures including:

1. Biological Plausibility:
Networks can be divided into two groups: those which employ biologically plausible learning rules (e.g. Hebbian learning) and those which employ learning rules that are not biologically plausible (e.g. back-propagation) (Thorpe and Imbert, 1989).

2. Information Flow:
Networks can be divided into feedforward and feedback networks. In feedforward networks, there are no feedback connections and information flows in the forward direction only. In feedback networks, feedback connections exist and information flows forwards and backwards.

3. Learning:
Networks can be divided into those trained with and without supervision.

4. Association:
Networks can be divided into auto-associative and hetero-associative networks. In auto-associative networks, the output is the same as the input. Hetero-associative networks associate an input pattern with a different output pattern.

5. Input:
Certain networks only accept binary inputs, whereas other networks operate on continuous valued inputs.

Various authors have devised a system of classification for neural networks.

Daniell (1991) suggests the factors which should be used to categorise ANNs are the way information flows through the network and the learning rules used.

Lippmann (1987) has devised a taxonomy for ANNs. As shown in Figure 2.8, the primary distinction is between networks using binary and continuous valued inputs. Below this, networks are divided into those trained with and without supervision.

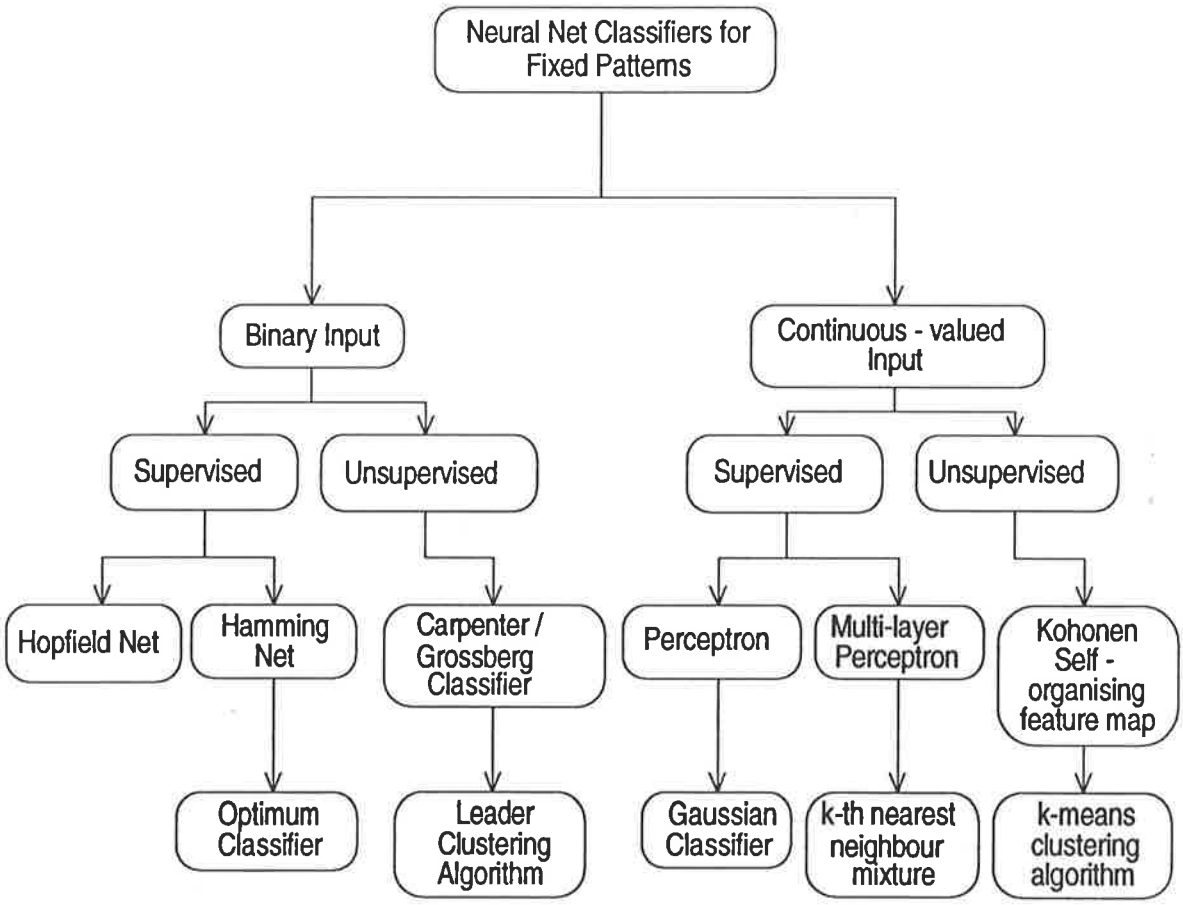


Figure 2.8: A Taxonomy of Neural Networks (Source: Lippmann, 1987)

Maren et al. (1990) have used structural similarity as the criterion for network classification. This classification has been adopted in this thesis and is summarised in Figure 2.9. Using this classification, networks are divided into six structurally related groups:

1. Multi-layered, feedforward networks.
2. Single-layer, laterally-connected networks.
3. Single-layer, topographically ordered networks.

4. Bi-layer, feed forward/feedback networks.
5. Multilayer cooperative/competitive networks.
6. Hybrid networks.

A brief description of these six groups of networks is given below. A more detailed discussion is given by Maier (1995).

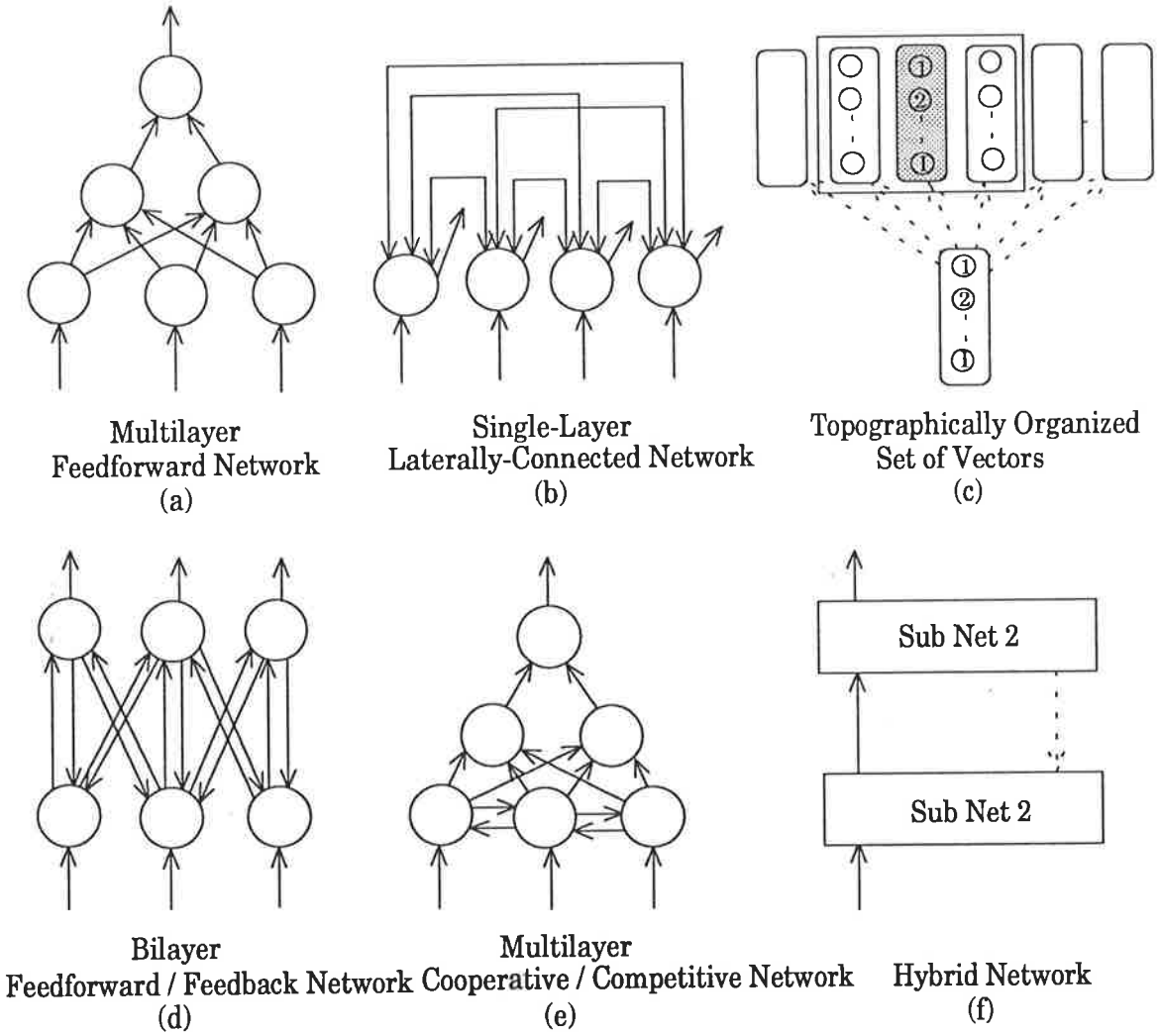


Figure 2.9: Six Types of Network Architectures (Source: Maren et al., 1990)

Multilayer feedforward networks have layered architectures, use supervised learning rules and only have feedforward connections (i.e. there are no lateral-, self-, or back-connections) (Maren et al., 1990). The difference between the various networks is in the way they learn. Perceptron networks (Rosenblatt, 1958), ADALINE and MADALINE networks (Widrow and Hoff, 1960) and back-propagation networks (Rumelhart et al., 1986) use calculus based learning rules, which are driven by the difference between the predicted and actual outputs. In contrast, the Boltzmann machine (Ackley et al., 1985) and directed random search networks (Matyas, 1965;

Solis and Wets, 1981) use random and statistical approaches to optimise the connection weights.

Single-layer, fully laterally connected networks act as auto-associators, as they can only store one pattern at a time. Hopfield networks (Hopfield, 1982) and brain-state-in-a-box networks (Anderson, 1983) belong to this class of networks. As with all recurrent networks, stability is a consideration, as information flows in more than one direction and oscillations can occur (Maren et al., 1990).

Vector-matching networks are single-layer networks that can act as auto-associators as well as optimisers. Unlike other network types, there are no connections between processing elements. Instead, pattern vectors are related to similar ones in a topological sense (Maren et al., 1990). The networks that belong to this class include learning vector quantisation (LVQ) networks (Kohonen, 1988; DeSieno, 1988) and self-organising maps (SOMs) (Kohonen, 1982).

Feedforward / feedback hetero-associative networks have both feed-forward and feedback connections, and may have lateral connections. They employ unsupervised learning rules and there is no distinction between the training phase and the operation phase (Maren et al., 1990). Adaptive resonance theory (ART) networks (described in Lippmann, 1987; Wassermann, 1989; Maren et al., 1990), bi-directional associative memory (BAM) networks (Kosko, 1987) and recirculation networks (Hinton, 1988) are examples of this type of network. As is the case with single-layer, fully laterally connected networks, oscillations can occur that may not converge to a stable state.

Multilayer cooperative / competitive networks are designed to mimic certain biological networks. The processing elements in the networks cooperate as well as compete with each other in order to perform the desired task. Usually, the cooperative and competitive processes are carried out by separate layers. This architecture results in subtle and sophisticated feature recognition and pattern recognition networks. The competitive mechanism is achieved by adding lateral, inhibitory connections to feed-forward networks (Maren et al., 1990). Examples of networks of this type include competitive learning networks (Rumelhart and Zipser, 1985), masking field networks (Grossberg, 1978), boundary contour systems (Grossberg and Mingolla, 1985a, 1985b), Hierarchical scene structures (Minsky and Maren, 1989) and the neocognitron (Fukushima, 1988).

Typically, hybrid networks combine a competitive, or unsupervised, layer with a supervised layer, resulting in rapid computation (Burke and Ignizio, 1992). This

architecture leads to two distinct phases in the learning process. The unsupervised learning component acts as a preprocessing unit which discovers features that are linear functions of the input vector. The results of this stage are passed to the supervised learning phase (Hrycej, 1990). Hybrid networks are usually used when known architectures are inadequate to perform the task at hand and are usually formed by combining networks of known architectures. Such networks can be strongly coupled or loosely coupled (Maren et al., 1990). Strongly coupled hybrid networks are treated as a single network. They consist of two or more networks that have been fused together to form a single structure in a way that highlights the strengths and minimises the weaknesses of the component networks. Weakly coupled hybrid networks are formed by linking two or more networks that retain their individual characteristics. Hamming networks (Lippmann, 1987), counter-propagation networks (Hecht-Nielsen, 1987a, 1988) and probabilistic neural networks (Specht and Shapiro, 1990) are examples of hybrid networks.

2.1.8 Back-Propagation Networks

The back-propagation network was developed by Rumelhart, Hinton and Williams (1986), and is the most widely used network today. Back-propagation networks (Figure 2.10) belong to the class of multilayer feedforward networks that use the delta learning rule.

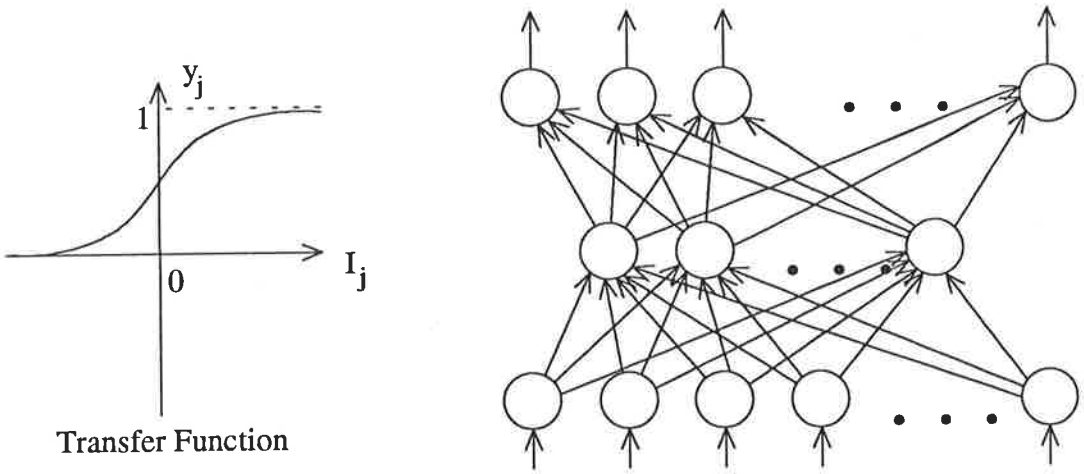


Figure 2.10: Typical Back-Propagation Architecture (Source: Maren et al., 1990)

Back-propagation networks are capable of adjusting the weights in the hidden layer(s) and hence solve the credit assignment problem formulated by Minsky and Papert in their book "Perceptrons" (Minsky and Papert, 1969). This is achieved by assigning part of the blame for erroneous outputs to all processing elements (NeuralWare, Inc., 1991).

Back-propagation networks are mapping networks that approximate mathematical functions or mappings from an input space to an output space using examples of the mapping's action. Mapping networks are variants of the methods of statistical regression analysis (Hecht-Nielsen, 1990). They can be designed and trained to map many complex patterns and have the ability to generalise with the aid of the hidden layer nodes, which perform nonlinear feature extractions (Maren et al., 1990).

Broomhead and Lowe (1988) suggest that back-propagation networks may be viewed as performing simple curve fitting operations in high dimensional space. In addition, learning is seen to be synonymous with producing a best fit surface in a high dimensional space to a finite set of data points (the training set), and generalising is seen to be equivalent to interpolating the test set on this fitting surface.

Back-propagation networks are fully interconnected. During recall, information flow is feed-forward only. The back-propagation architecture is the same as that for multilayer perceptrons with the following exceptions (Maren et al., 1990):

1. Back-propagation networks use continuous transfer functions, such as the sigmoid function. These functions are asymptotic for infinitely large and small values of activation. In fact, for most input values, the output of the transfer function takes on one of the two asymptotic values, allowing the output to be roughly grouped into one of two classes.

   During training, weights are adjusted in proportion to the derivative of the sigmoid function. As shown in Figure 2.11, the derivative of the sigmoid function is always positive, is close to zero for extreme input values and is at its maximum when the input is zero. As a result, small weight changes occur when the output is either zero or one, and large weight changes occur when the output is in the middle range. This is desirable, as outputs of zero or one represent stable states and no further changes are required. If the output is in the medium range, however, large changes are required to drive the output to one of the two stable states.

2. In contrast to perceptron networks, back-propagation networks use the generalised delta learning rule discussed in Section 2.1.8.1 below.

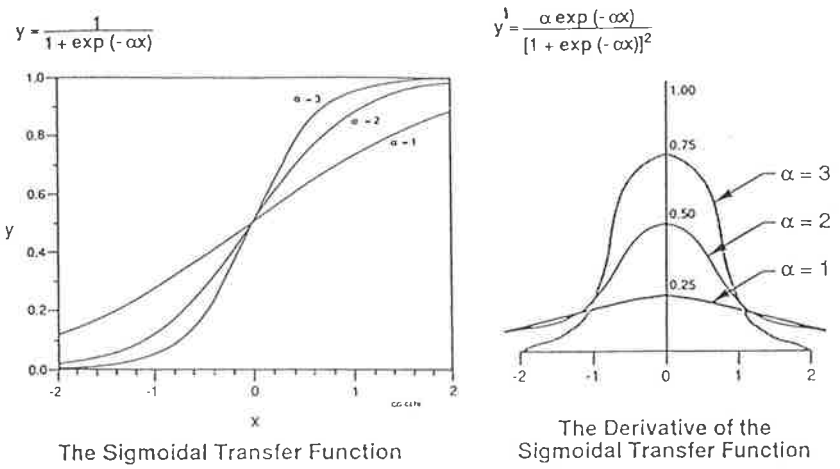


Figure 2.11: The Sigmoidal Transfer Function and its Derivative (Source: Maren et al., 1990)

2.1.8.1 Learning

The most interesting aspect of the back-propagation network, and the reason for its success and popularity, is the back-propagation learning algorithm, which is based on delta rule learning for networks with no hidden nodes.

The delta rule

Delta rule learning is a supervised learning algorithm for networks without hidden nodes. The input is presented at the input layer and passed to the output layer. The predicted output is then compared with the desired output, and the difference, or delta, between the predicted and the desired response of the network is calculated. The weights are then changed in accordance with the following equation (Jones and Hoskins, 1987):

$$\Delta w_{ji} = \eta \, (o_k^d - o_k^p) \, I_i \qquad (2.13)$$

where
o_k^d = the desired (actual, historical) output at node k
o_k^p = the predicted output at node k
Δw_{ji} = the change in weight between nodes i and j
I_i = the activation of the input node
η = the learning rate

This basically means that the credit (or blame) for the error assigned to the input units is proportional to their activation. The more active nodes are, the more they are responsible for the actual output, and hence the error produced. As a result, the weights of the most active nodes undergo the largest modification. In fact, the direction of weight change is such that it will minimise the sum of the squares of the output deltas. As learning progresses, the network cycles through the training set until a specified performance level is reached.

The delta rule works well for single layer systems but cannot be used for multilayer systems, as there is no mechanism for working out weight changes for nodes in the hidden layer(s). This problem was solved by the introduction of the back-propagation or generalised delta rule.

The generalised delta rule

The generalised delta rule is a supervised learning rule for adjusting weights in multilayer networks. It is widely used and is discussed by many authors (Maren et al., 1990; Hecht-Nielsen, 1990; Lippmann, 1987; Burke and Ignizio, 1992; White, 1990; Jones and Hoskins, 1987; Minai and Williams, 1990; NeuralWare, Inc., 1991; Cheung et al., 1990).

The goal of the learning procedure is to find a set of weights that enable the network to perform the desired input/output mapping. During training, the mean squared difference between the predicted and the desired output is calculated as a function of the weights. A different error is obtained for each combination of weights, creating a mean squared error surface sitting above the weight space (Figure 2.12). This error surface has a large number of local minima, because of the large number of combinatorial permutations of the weights that leave the network input/output function unchanged. The aim of the training procedure is to find a set of weights that will minimise the error function.

The weight changes are obtained by performing gradient descent in weight space. Each weight is changed in proportion to the value of the partial derivative of the error function with respect to that weight. Consequently, weight changes are made down the gradient of the error surface.

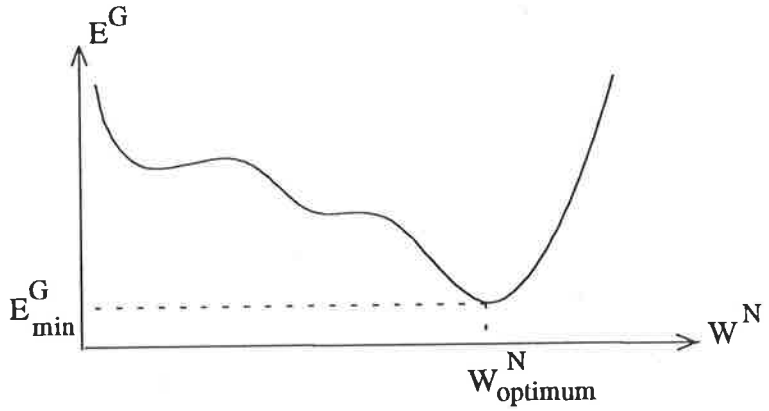


Figure 2.12: A Typical Error Surface (Source: Hecht-Nielsen, 1990)

The weights are adjusted in two stages. During the first stage, the weights between the hidden layer and the output layer are adjusted. In the second stage, the weights between the hidden layer and the input layer are adjusted. An iterative method is used, which propagates the error terms needed for weight adjustment back from the output layer to the input layer via the hidden layer(s). As training patterns are presented to the network, and weights are adjusted gradually in order to approximate the correct input/output mapping, the network goes through the following stages of learning (Cheung et al, 1990):

1. Error-Convergent Stage:
In this stage, the errors produced by the various patterns tend to converge towards their mean.

2. Competition Stage:
The competition stage immediately follows on from the error-convergent stage. Initially, the gradients of the individual patterns are in opposing directions and tend to cancel each other out, resulting in an overall gradient equal to zero. Consequently, the individual pattern gradients compete with each other in order to drive the overall gradient away from zero. The changes in the overall gradient will be such that they result in a decrease in the error function. This will cause the errors to eventually move away from their mean.

3. Domination Stage:
In this stage, some patterns are dominantly trained. The errors of these patterns decrease relatively quickly in comparison with other patterns. As learning progresses, this domination situation disappears.

In general, stage one is very short, whereas the duration of stages two and three is highly problem dependent.

The detailed back-propagation algorithm is given below (NeuralWare, Inc., 1991):

1. • The input vector, **i**, is presented at the input layer, and the desired output, $\mathbf{o}^d$, is applied at the output layer.

   • The input vector is then propagated through the network producing the predicted output, $\mathbf{o}^p$.

   • As the input propagates through the layers, the output state is obtained for each node as a function of the input activation, I.

$$y_j^{[s]} = F\left(\sum_i \left(w_{ji}^{[s]} x_i^{[s-1]} \right)\right) = F\left(I_j^{[s]}\right) \qquad (2.14)$$

   where $x_j^{[s-1]}$ = the current input from the ith processing element in layer [s-1]

   $y_j^{[s]}$ = the current output state of the jth processing element in layer [s]

   $w_{ji}^{[s]}$ = the weight on the connection joining the ith processing element in layer [s-1] to the jth processing element in layer [s]

   $I_j^{[s]}$ = the weighted summation of inputs to the jth processing element in layer [s]

   • Traditionally, F is the sigmoid function, but it can be any squashing function.

2. • The scaled error and the delta weight are calculated for each processing element in the **output layer** in accordance with the following equations.

   • The global error function is given by:

$$E^G = \frac{1}{2} \sum_{k=1}^{N^O} \left((o_k^d - o_k^p)^2 \right) \qquad (2.15)$$

   where E^G = the global error

o_k^d = the desired output for the kth processing element
o_k^p = the predicted output for the kth processing element

- The scaled local error at each processing element in the output layer, E^L, is given by:

$$E^{L\,[o]} = (o_k^d - o_k^p)\ F'(I_k) \tag{2.16}$$

where $F'(I_k)$ = the derivative of the transfer function with respect to its input

- The global error, E^G, is minimised by modifying the weights using the following gradient descent rule:

$$\Delta w_{ji}^{[s]} = -\eta \frac{\partial E^G}{\partial w_{ji}^{[s]}} \tag{2.17}$$

where η = the learning rate

In other words, each weight is changed in proportion to the size and direction of the negative gradient of the error surface.

- By rearranging and substituting, the following expression for the weight change is obtained:

$$\Delta w_{ji}^{[s]} = \eta\ E^{L}{}_j^{[s]}\ x_i^{[s-1]} \tag{2.18}$$

where $E^{L}{}_j^{[s]}$ = the scaled local error in the jth processing element in layer [s]

3. • The scaled error and the delta weight are calculated for each processing element in the layers between the layer below the output layer and the layer above the input layer.

- The local error that is passed back through the layers is given by:

$$E^{L}{}_j^{[s]} = -\frac{\partial E^G}{\partial I_j^{[s]}} \tag{2.19}$$

- The relationship between the local error at a particular processing element at level [s] and all the local errors at level [s+1] is obtained by applying the chain rule twice to Equation (2.19).

$$E^{L}{}_{j}{}^{[s]} = F'(I_j^{[s]}) \sum_k (E^{L}{}_{k}{}^{[s+1]} w_{kj}{}^{[s+1]}) \qquad (2.20)$$

- The weight update equation is the same as that for the output layer and is given by Equation (2.18).

4. • All the weights in the network are updated by adding the delta weights to the corresponding previous weights.

Once training is complete, the weights are frozen. Training is the only time data is back-propagated through the network. During recall, the network is strictly feed-forward.

Use of training data (Hecht-Nielsen, 1990; Specht and Shapiro, 1990)

The training data consist of examples of the desired input/output mapping. Generally the training data are divided into two subsets; one portion for training and one portion for testing the performance of the network. It is vital that the test data are not used during training. When a limited data set is available, the holdout method can be used (Specht and Shapiro, 1990). When the holdout method is used, one pattern is held out and training is carried out using the remaining patterns. The held out pattern is then tested against the trained network. The above procedure is repeated, holding out each pattern in turn.

Generally, the more training examples are presented to the network, the better it performs the desired input/output function. However, back-propagation networks can "overtrain". When too many training examples are presented to the network, it tends to learn the specific data set better than the general problem. This is undesirable, as the network should have the ability to generalise. When a network overtrains, training should be stopped. However, it is difficult to know when to stop training. The only way to determine whether a network is overtraining is to stop training at periodic intervals, freeze the weights temporarily and test the performance of the network using the test set. As shown in Figure 2.13, the mean squared error will decrease in the initial stages of training. As the network starts to overtrain, the error will increase again.

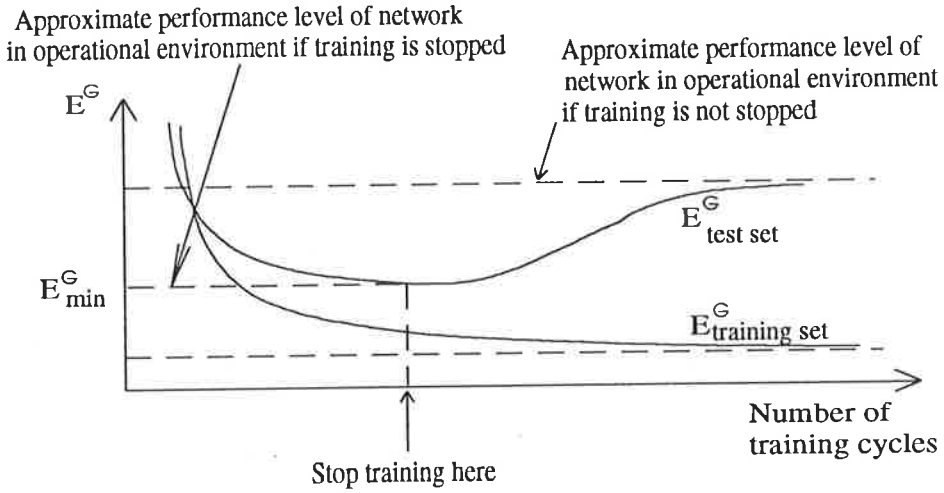


Figure 2.13: The Phenomenon of Overtraining (Source: Hecht-Nielsen, 1990)

Stopping criteria

Stopping criteria determine when the training process should be stopped. Examples of stopping criteria include the following (Patterson, 1993; Lachtermacher, 1993; NeuralWare, Inc., 1991):

1. Learning can be stopped when the magnitude of the derivative of the transfer function reaches a sufficiently small value for all connections.
2. Learning can be stopped after a fixed number of training samples have been presented to the network.
3. Learning can be stopped when there is no further improvement in network performance during the testing phase. This involves stopping training at regular intervals, presenting the test set to the network and comparing network predictions with the actual values. This method is known as cross-validation.
4. Learning can be stopped when the value of the error function reaches a sufficiently small value during training. However, this can lead to overtraining.

2.1.8.2 Measuring Network Performance

When the performance of a network is assessed, two factors should be taken into account: learning speed and generalisation ability. It has been shown that learning speed and generalisation ability are closely related (LeCun, 1989). Ultimately, it is the network's ability to generalise that is most important, as it determines the amount of data needed to train the network. The network's generalisation ability is a measure of

how well the network has learned to approximate the desired mapping for arbitrary inputs.

A means for assessing generalisation ability is to calculate the root mean squared error (RMSE) between the predicted and the desired output. Using the RMSE as a performance measure has the advantage that large errors receive more attention than small ones. The RMSE is also more sensitive to errors on frequent inputs (Hecht-Nielsen, 1990). Alternative methods for assessing a network's generalisation ability include the average absolute error (AAE), the average absolute percentage error (AAPE), the maximum absolute error (MAE) and the median squared error (MSE) (Hecht-Nielsen, 1990).

The RMSE, AAPE and the AAE are calculated as follows. If there are P^{TE} patterns in the testing set:

$$\text{RMSE} = \left\{ \frac{1}{P^{TE}} \sum_{k=1}^{P^{TE}} (o_k^d - o_k^p)^2 \right\}^{\frac{1}{2}} \tag{2.21}$$

$$\text{AAPE (\%)} = \frac{100}{P^{TE}} \sum_{k=1}^{P^{TE}} \frac{\left| o_k^d - o_k^p \right|}{o_k^d} \tag{2.22}$$

$$\text{AAE} = \frac{1}{P^{TE}} \sum_{k=1}^{P^{TE}} \left| o_k^d - o_k^p \right| \tag{2.23}$$

2.1.8.3 Network Design

As discussed in Section 2.1.8.2, the ability to generalise is the most important network performance measure. The likelihood of correct generalisation depends on the size of the hypothesis space (the total number of networks being considered), the size of the solution space (the number of networks that have good generalisation ability) and the number of training examples (LeCun, 1989). If the hypothesis space is too large, or the number of training examples is too small, then there exist many networks consistent with the training data, but only very few of these lie in the true solution space, resulting in poor generalisation ability. Conversely, in order to obtain good generalisation ability with a network of increased generality, more training examples are required. However, it should be noted that Wann et al. (1990) suggest that increasing the number of training samples does not guarantee better generalisation ability. Hertz et al. (1991) indicate that good generalisation ability can be achieved by building as much a priori

knowledge as possible into the network, and by minimising the number of free network parameters.

In an attempt to improve network performance, it is important to optimise the network geometry, the initial weight distribution and the learning rate.

Network Geometry

Several authors give guidelines as to how many hidden layers should be used (Maren et al., 1990; Hertz et al., 1991).

As shown by Cybenko (1988), at most two hidden layers are needed to approximate a particular function to a given accuracy, provided that there are sufficient processing elements per layer. The reason why two hidden layers are enough to approximate any function is based on the argument that any "reasonable" function can be represented by a linear combination of localised "bumps" and that such "bumps" can be constructed with two hidden layers. However, it may be possible that a network using more than two hidden layers may find a solution with less units in total or may speed up learning. Another point to consider is that although some functions may be represented using an architecture with two hidden layers, they may not be capable of learning with that architecture.

As a general guide, one hidden layer is probably sufficient for classification (decision boundary) problems, where the output node with the greatest activation will determine the category of the input pattern. However, if the outputs need to be continuous functions of the inputs, two hidden layers, or different transfer functions, such as radial basis functions (Section 2.1.8.5), should be used (Maren et al., 1990).

When choosing the number of hidden layer nodes, the following should be kept in mind (Maren et al., 1990):

1. The hidden nodes represent features in the input space and not the exact input pattern.
2. Keeping the number of hidden layer nodes to a minimum reduces the number of weights that need to be adjusted, and hence reduces the computational time needed for training.

Hidden nodes detect relationships between input and output values. If there is an insufficient number of hidden nodes, it may be difficult to obtain convergence during training, as the network is unable to create adequately complex decision boundaries. If

too many hidden nodes are used, the network may lose its ability to generalise. This is similar to the area of statistics and curve fitting, where too many free parameters result in overfitting, as shown in Figure 2.14.

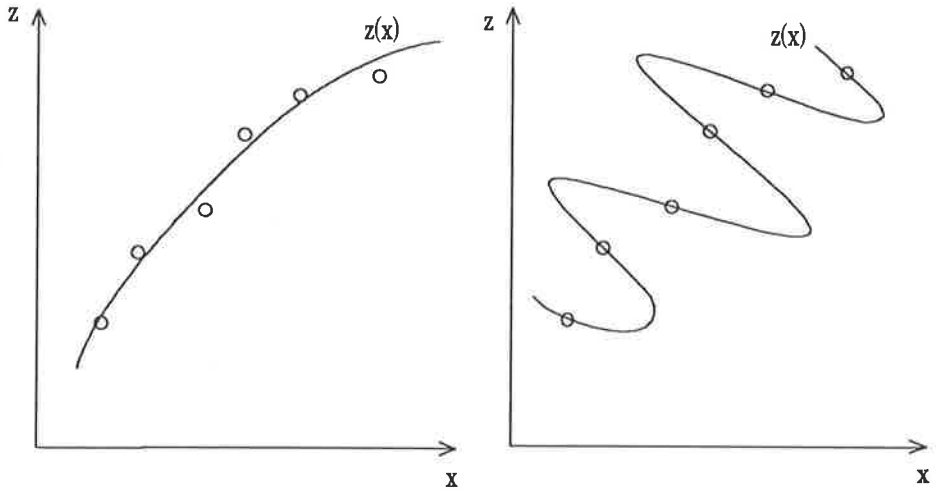


Figure 2.14: The Phenomenon of Overfitting (Source: Hertz et al., 1991)

A curve fitted with too many parameters follows all the small details or noise but is unable to interpolate or extrapolate effectively. The same applies to artificial neural networks; too many nodes result in poor generalisation.

Guidance for choosing the correct number of hidden layer nodes is given by various authors, some of which are a function of the number of network inputs and outputs, whereas others depend on the number of training samples available.

- Hush and Horne (1993) suggest that the number of independent training samples should be approximately ten times the number of weights for the network to have good generalisation ability.

- Baum and Haussler (1989) derived a theoretical relationship between the number of training samples and the network size required to achieve valid generalisation. They suggest that for a given network geometry, the number of training samples required, P^{TR}, is given by:

$$P^{TR} \geq N^{W} / \alpha \qquad (2.24)$$

where N^{W} = the number of weights
and $0 < \alpha \leq 1/8$

If the above relationship is satisfied, the fraction of test samples that will be correctly classified is $(1 - \alpha)$.

- Rogers and Dowla (1994) suggest using the rule of thumb that the number of training samples (P^{TR}) should be greater or equal to the number of weights in the network (N^{W}).

- Weigend et al. (1990, 1991) suggest using the following heuristic rule for determining the initial number of hidden nodes:

$$\frac{1.1\, P^{TR}}{10} \leq N^{H1} (N^{I} + 1) < \frac{3\, P^{TR}}{10} \tag{2.25}$$

where P^{TR} = the number of training samples
N^{H1} = the number of hidden nodes
N^{I} = the number of inputs

However, Chakraborty et al. (1992) found this rule to be too restrictive when limited training data are available.

- NeuralWare, Inc. (1991) give the following upper bound for the number of hidden layer nodes:

$$N^{H1} \leq \frac{P^{TR}}{\alpha\, (N^{I} + N^{O})} \tag{2.26}$$

where P^{TR} = the number of training samples
N^{O} = the number of outputs
N^{I} = the number of inputs
N^{H1} = the number of hidden nodes
α = a constant

The constant, α, ranges between 2 and 10; 2 is used for very clean data while 10 is used for noisy data.

- Hecht-Nielsen (1987b) gives the upper limit for the number of hidden nodes as $2N^{I}+1$ for a network with one hidden layer.

- Maren et al. (1990) report that for many applications, the optimum number of hidden layer nodes has been found to be less than the number of inputs.

- DeSilets et al. (1992) specified the number of hidden layer nodes as a percentage of the total number of nodes in the input and output layers. They found that the optimum percentage ranged from 20% to 50%.

Kudrycki (1988) empirically determined that for networks with two hidden layers, the optimum ratio of first to second hidden layer nodes is 3:1.

When limited data sets are available, the following methods may be used to increase generalisation ability (LeCun, 1989):

1. Several connections can be controlled by a single parameter by using a technique called "weight sharing".
2. "Useless" connections can be dynamically deleted during training.

Methods for determining the optimum network architecture include the following:

- Harp et al. (1989) applied genetic algorithms to populations of descriptions of back-propagation networks. Each description (genome) contains values of various network parameters (genes). The population converges to a near optimal architecture by using a technique that ensures that good building blocks found in one trial are refined and combined with good building blocks from other trials.

- Fahlman and Lebiere (1990) developed an algorithm which adds individual nodes as needed during training. Initially, a network without hidden layers is trained to optimum performance. Subsequently, hidden nodes are added and trained one at a time until the model performance reaches a target value.

- Weigend et al. (1990) added a term to the error function that penalises large networks, resulting in the removal of excess weights during training.

- Sietsma and Dow (1991) and Chung and Lee (1992) developed pruning methods for removing superfluous processing elements based on the examination of the node outputs.

Initial weight distribution

The size and distribution of the connection weights at the beginning of training is very important. If the weights are identical, the network may not learn (Rumelhart et al., 1986). If the magnitude of the initial connection weights is too large, the activation function will be large and the transfer function might saturate, leaving the network in a local minimum or in an undesirable region of the weight space. Consequently, the initial weights should be small and randomly distributed.

Learning rate (DeSilets et al., 1992; Minai and Williams, 1990; White, 1990)

The learning rate determines the absolute size of the weight change during learning. It determines the magnitude of the steps taken down the error surface. The learning rate is one of the factors which determines how fast the network learns.

Error surfaces have a multitude of areas with shallow slopes. As a result, large learning rate values are needed for rapid learning. However, rapid training may not necessarily be effective. If the learning rate is too large, the network might jump to undesirable areas in weight space.

Generally, the learning rate has to be found by trial and error. Rumelhart et al. (1986) have reported successfully using values ranging from 0.05 to 0.75 for a variety of problems.

2.1.8.4 Deficiencies of the Back-Propagation Algorithm

The deficiencies of the back-propagation algorithm have been widely discussed (Cheung et al., 1990; Hrycej, 1990; Specht and Shapiro, 1990; Burke, 1991; Vemuri, 1988; Burke and Ignizio, 1992; Jones et al., 1990).

Deficiencies of back-propagation networks include that they:

1. Take a long time to train.
2. May suffer from "overtraining", leading to poor generalisation.
3. Can be temporally unstable.
4. Rely on non-local computation.
5. Require re-training if new information arises.
6. Tend to get stuck in local minima of the error surface, and hence global minima cannot be guaranteed.

7. Assume all patterns are equally effective during training.
8. May suffer from "network paralysis", which occurs when some weights actually freeze and the network stops adjusting these.
9. Require a great deal of training data.
10. Are sensitive to the value of control parameters such as the learning rate.

Some of the deficiencies related to network training stem from the fact that the assignment of optimal weights is a high dimensional, nonlinear optimisation problem for which there is no adequate solution algorithm.

2.1.8.5 Ways of Improving the Back-Propagation Algorithm

There have been a number of attempts to address the deficiencies of the back-propagation algorithm discussed in Section 2.1.8.4, some of which are discussed below.

Momentum (NeuralWare, Inc., 1991; Minai and Williams, 1990)

As discussed in Section 2.1.8.3, it is difficult to determine an appropriate learning rate. At points of high curvature, a small learning rate is required to avoid divergent behaviour. At points of shallow slopes, a large learning rate is required in order to avoid slow learning. To resolve this contradiction, the momentum term was introduced.

The inclusion of the momentum term modifies Equation 2.18 as follows (NeuralWare, Inc., 1991):

$$\Delta w_{ji}^{[s]}(t) = \eta \; E^{L}{}_{j}^{[s]} \; x_i^{[s-1]} + \mu \, \Delta w_{ji}^{[s]}(t-1) \qquad (2.27)$$

where μ = the momentum value

The fact that a function of the previous weight change is passed through to the current weight change ensures that general trends are reinforced and oscillatory behaviour cancels out. A positive momentum value provides a built-in inertia, allowing a small learning rate but faster learning.

Modifying the derivative (NeuralWare, Inc., 1991)

Modifying the derivative of the sigmoid transfer function can nearly double the learning speed. When the node activation value becomes saturated (0.0 or 1.0), the derivative, and hence the scaled error, becomes 0.0 and the network ceases to learn. This problem is eliminated by adding a small positive offset to the derivative.

Different error functions (NeuralWare, Inc., 1991; Bishop, 1990)

As discussed in Section 2.1.8.3, if the number of hidden nodes is insufficient, the desired accuracy may not be achieved. However, if there are too many nodes, the network overfits the data as shown in Figure 2.14.

The following error function has been proposed by Bishop (1990) to smooth out the output function and to eliminate the problem of overfitting (Figure 2.14).

$$E^G = \frac{1}{2} \sum_k ((o_k^d - o_k^p)^2) + c\,E^c \qquad (2.28)$$

where c = a constant

$$E^c = \frac{1}{2} \int_a^b \kappa^2 \, dx = \frac{1}{2} \int_a^b \frac{(z'')^2}{(1 + (z')^2)^3} \, dx$$

κ = the curvature of the line $z = z(x)$ - see Figure 2.14.

By using Equation 2.28, many hidden nodes can be used without overfitting the data.

Many different error functions can be used. The only requirement is that their derivatives can be calculated at the output layer. Examples of possible error functions include the cubic and the quartic error functions shown below.

$$E^G = \frac{1}{3} \sum_k \left|(o_k^d - o_k^p)^3\right| \qquad (2.29)$$

$$E^G = \frac{1}{4} \sum_k (o_k^d - o_k^p)^4 \qquad (2.30)$$

Cumulative update of weights (NeuralWare, Inc., 1991)

Learning rules determine how the weight updates are carried out during training. When a cumulative delta learning rule is used, a number of training samples are presented to the network before the weight update is carried out. The number of training samples presented to the network before each weight update is called the epoch size (ε). The epoch size used can be equal to the entire training set or a subset of the training set. By presenting a number of training samples to the network before the weights are updated, rather than updating the weights after the presentation of each training sample, learning speed can be increased, as each update works more towards reaching the global minimum in the error surface, rather than the local minimum for the particular pattern being considered (Neuralware, Inc., 1991). However, this advantage may be lost if the epoch size is too large, as many more calculations need to be carried out for a single update.

Fast back-propagation (NeuralWare, Inc., 1991)

Fast back-propagation is a variation of the back-propagation algorithm developed by Samad (1988). In fast back-propagation, the weight update equation is given by:

$$\Delta w_{ji}^{[s]} = \eta \; E^{L}{}_{j}{}^{[s]} \; (x_i^{[s-1]} + E^{L}{}_{i}{}^{[s-1]}) \qquad (2.31)$$

Using this variation of Equation 2.18 can dramatically reduce the number of iterations required for convergence.

Dynamic learning rate (Schmidhuber, 1989; White, 1990)

Another way to speed up the learning process is to use a dynamically varying learning rate. Gradient descent methods attempt to find a point of zero gradient in the error surface, corresponding to a local minimum in the error. However, it would be more appropriate to find the zero points of the error function. This can be achieved by using a dynamically varying learning rate that is computed during the presentation of each training sample.

The learning rate is selected so that the updated weight vector points to the intersection of the weight hyperplane and the line defined by the current error and the current gradient. It is assumed that the error function can be locally approximated by its tangential hyperplane. Thus the gradient is used for linear extrapolation in order to update the weight vector, producing a zero value for the error function or a value close

to zero. The above process is repeated with the updated vector as the starting point for the next iteration. One step of the process is illustrated in Figure 2.15 for a two-dimensional example.

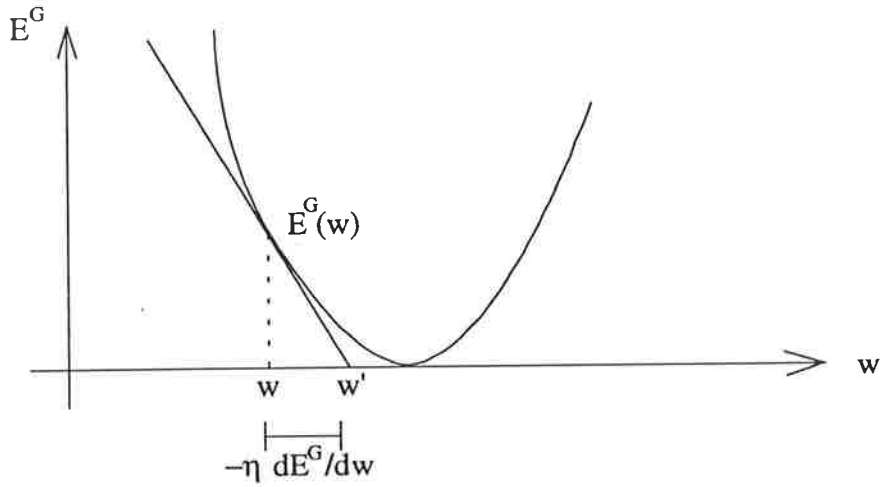


Figure 2.15: Illustration of One Iteration Step (Source: Schmidhuber, 1989)

The value of the learning rate for each training pair is given by (Schmidhuber, 1989):

$$\eta = \frac{E^G}{\sum_k \left(\frac{\partial E^G}{\partial w_k} \right)^2} \qquad (2.32)$$

When the denominator of Equation 2.32 is equal to zero, the learning rate is set equal to zero.

The above method is based on the Newton-Raphson method for finding the roots of an equation. As the learning rate is calculated automatically, the need for parameter fine tuning and the empirical uncertainty in calculating the learning rate are eliminated. A similar method is described by White (1990).

Delta-Bar-Delta algorithm (NeuralWare, Inc., 1991; Minai and Williams, 1990)

The delta-bar-delta (dbd) algorithm was developed by Jacobs (1988) in an attempt to increase the convergence speed of back-propagation networks. The slope of the back-propagation error surface varies considerably along different weight directions. By using a constant learning rate, and hence step size, for each weight, the distance moved down the error surface will be considerably larger for an area with a steep slope than

one that has a shallow slope. As a result, local minima across steep slopes can be jumped over and descent along shallow slopes is very slow.

The delta-bar-delta algorithm addresses the above problem by using the following four heuristics (Minai and Williams, 1990):

1. All weights have individual learning rates.
2. The weights are varied in response to changes in the geometry of the error surface.
3. When the sign of the error surface gradient remains unchanged for several iterations, the corresponding learning rate is increased, as this indicates that a minimum lies ahead.
4. When the sign of the error surface gradient changes for several consecutive time steps, the learning rate is decreased, as this indicates that a minimum is being jumped over.

Although the delta-bar-delta algorithm speeds up the learning process considerably, it suffers from the following drawbacks (Minai and Williams, 1990):

1. Using the momentum term in conjunction with the delta-bar-delta algorithm can result in divergent behaviour.
2. The learning rate can sometimes increase to such an extent that the decrease described in 4 above might not be sufficient to prevent wild jumps.

The extended-delta-bar-delta (edbd) algorithm was developed (Minai and Williams, 1990) to overcome these shortcomings. The following changes were made to the delta-bar-delta algorithm:

1. The increase in the learning rate is changed from a constant to an exponentially decreasing function. As a result, the learning rate increases faster on flat areas than on areas of greater slope.
2. A time-varying momentum term is added to each weight. The momentum term is varied in the same way as the learning rate, resulting in an increased momentum term on plateaux and an exponentially decreasing momentum term in the vicinity of local minima.
3. A ceiling is placed on the learning rate and momentum values in a further attempt to prevent wild jumps and oscillations in weight space.
4. A memory with recovery feature is built into the algorithm. It has the function of saving the best result seen so far, and restarting the search at this point with

attenuated learning rate and momentum values when the error becomes greater than a specified value.

Adaptive back-propagation (Silva and Almeida, 1990)

The back-propagation algorithm performs weight updates in the direction that yields the maximum error reduction. However, there is good reason to do exactly the opposite. If the gradient is small, small steps are actually taken, although the error surface has a gentle slope so that bigger steps are more appropriate. If the gradient has a large value, the back-propagation algorithm takes large steps whereas small steps are required to avoid oscillations as a local minimum is being approached.

Adaptive back-propagation makes use of the same principles as the delta-bar-delta algorithm and the extended-delta-bar-delta algorithms. The adaptive back-propagation algorithm adjusts the learning rate as learning progresses. If the sign of a certain component of the gradient remains the same for several iterations, the error surface has a smooth variation along this axis and an increase in the learning rate is warranted. If the sign of a component changes in several consecutive iterations, the learning rate is decreased to avoid oscillations. The adaption of the learning rate occurs at each epoch and is exponential in nature. The optimal learning rate values are usually determined in a few learning cycles, resulting in a rapid reduction of the total output error.

Dynamic training sets and weighting factors

Back-propagation training can be slowed down by some "poorly-trained patterns" (Cheung et al, 1990). Such patterns usually belong to infrequent classes, which form part of random training sets. Generally, frequent patterns will be dominantly trained, leaving the information of the infrequent patterns uncaptured. The back-propagation algorithm does not make any concessions for poorly trained patterns. However, two simple modifications can be made to the back-propagation algorithm to take poorly trained patterns into account (Cheung et al., 1990):

1. Using a Dynamic Training Set

The training set can be enlarged by including more examples of infrequent classes. Poorly trained patterns can be identified, as they produce the largest errors. If poorly trained patterns remain a feature, the size of the training set should be increased further. On the other hand, when the behaviour of the error function normalises, the

training set can be reduced. The training set is thus a dynamic one, as the size and content of the set is changing as part of training.

2. Using Weighting Factors

The direction in which the system is moving down the error surface should incorporate weighting factors to take into account the relative errors of training patterns.

Functional link networks (NeuralWare, Inc., 1991)

Functional link networks differ from traditional back-propagation networks in that they have additional nodes in the input layer that may dramatically increase the learning rate. The additional nodes can be added in one of two ways:

1. Outer Tensor Model

In this model, the additional inputs are created by multiplying each component of the input vector with the entire input pattern. For example, if the input vector is given by $[i_j : 1 \leq j \leq N^I]$, then the inputs for the tensor model are given by $[i_j, i_j i_i : 1 \leq j \leq N^I, j \leq i]$. In this way, no new information is added, but the representation of the input space is enhanced, making the learning process less difficult.

2. Functional Expansion Model

In this model, the additional inputs consist of functions of the existing inputs. For example, in addition to i_j, there might be input nodes for $\sin(\pi i_j)$, $\sin(2\pi i_j)$. Again, the representation of the input space is enhanced by mapping the input vectors into a larger pattern space.

Supplementary learning (Kimoto and Asakawa, 1990)

Supplementary learning was developed to cater for high-speed learning with a large volume of data. This technique has two important features:

1. Prior to commencement of training, tolerances are set for all output units. As the network learns, only errors exceeding this tolerance are propagated back through the network. As a result, the computational load drops as learning progresses because fewer errors will exceed the tolerance.

2. The learning rate parameter (η/P^{TR}) is adjusted automatically, and is dependent on the size of the training set. The weight update equation is given by:

$$\Delta w(t) = \frac{\eta}{P^{TR}} \frac{\partial E^G}{\partial w} \mu \Delta w(t-1) \qquad (2.33)$$

where P^{TR} = the number of training patterns used

As the learning rate is divided by the number of data items that actually require error back-propagation, its value does not need to be changed during the course of training. Initially, the learning rate parameter is relatively small, as the number of learning patterns is large. As learning progresses, the learning rate parameter is increased automatically because the number of learning patterns decreases.

The result of the above measures is that the supplementary learning procedure is very fast and that there is no need to adjust the learning parameter.

Radial basis functions

The use of radial basis functions is discussed by a number of authors (Hertz et al., 1991; Maren et al., 1990; Broomhead and Lowe, 1988; Jones et al., 1990).

The performance of back-propagation networks can be improved by using radial basis functions as the transfer function in the hidden layer (Broomhead and Lowe, 1988). Each processing element in the hidden layer represents a radial basis function centre. The response of the hidden layer nodes is given by the following function (Jones et al., 1990):

$$z(I_j) = \sum_{j=1}^{N^{H1}} \lambda_j \, f^{rb}{}_j (I_j) \qquad (2.34)$$

where λ_j = a coefficient

N^{H1} = the number of basis functions (the number of hidden nodes)

$f^{rb}{}_j(.)$ = a radial basis function at node j

A common radial basis function is the normalised Gaussian function given below (Hertz et al., 1991):

$$f^{rb}_j (I_j) = \frac{\exp\left(\frac{(I_j - c_j)^2}{2\sigma_j^2}\right)}{\sum_k \exp\left(\frac{(I_k - c_k)^2}{2\sigma_k^2}\right)} \qquad (2.35)$$

where c_j = the centre of the jth radial basis function
σ_j = the variance or spread around the centre, c_j, of the jth function
I_j = the input to the jth neuron

An example of a radial basis function is given in Figure 2.16.

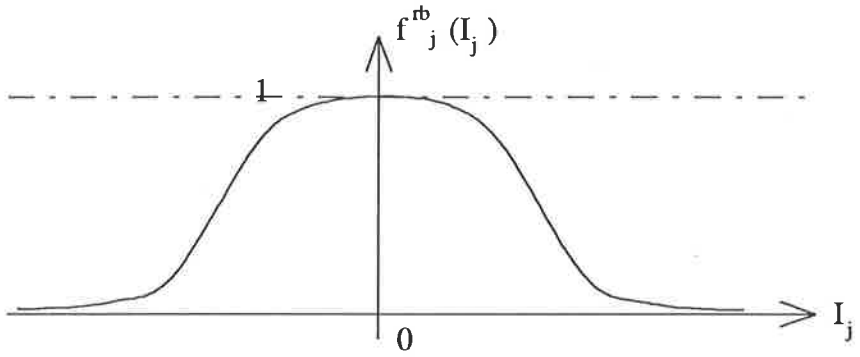


Figure 2.16: An Example of a Radial Basis Transfer Function
(Source: Maren et al., 1990)

The use of radial basis functions can be advantageous when the network is required to perform continuous function mappings. When radial basis functions are used, the network only requires one hidden layer, as each unit has its own receptive field in the input space; a region centred on c_j (i.e. on the centre of each hidden layer processing element) with size proportional to σ_j. The goal is to pave the input space with these receptive fields. In order to achieve that goal, appropriate receptive field centres, c_j, and widths, σ_j, need to be determined.

Hertz at al. (1991) suggest that vector quantisation techniques, such as the competitive learning algorithm, can be used to obtain suitable receptive field centres. The problem of how to adaptively set the radial basis function centres has also been addressed by Moody and Darken (1989) and Saha and Keeler (1990). As the performance of the network is not very sensitive to the variance, it is usually determined by an ad hoc choice, such as the mean distance to the neighbouring centres (Hertz et al., 1991).

Once the input space is covered by appropriate receptive fields, the problem is almost solved. If a particular input lies in the middle of a receptive field, only the

corresponding processing element is activated, producing the network output. If a particular input lies somewhere between two receptive field centres, the two processing elements involved are activated to a certain degree, and the network output will be a weighted average of the individual node outputs. As a result, the network provides a sensible, smooth fit to the desired function.

Once the radial basis function centres have been defined, the connection weights from the input layer to the hidden layer nodes are effectively fixed. Consequently, only the weights from the hidden layer nodes to the nodes in the output layer need to be determined during training. These connection weights correspond to the λ_j coefficients in Equation 2.34, and can be obtained by linear optimisation. For example, the delta rule can be used to adjust the connection weights, which will minimise the mean squared error between the actual and desired outputs. As a result of this linearity, the learning speed of radial basis function networks is higher than that of conventional back-propagation networks.

One problem associated with radial basis function nets is that they do not lend themselves to electronic implementation (Hertz et al., 1991).

2.1.9 Applications

Traditionally, ANNs have been used to carry out cognitive tasks performed naturally by the brain, including recognising a familiar face, learning to speak and understand a natural language, identifying handwritten characters, retrieving contextually appropriate information from memory and determining that a target seen from different angles is in fact the same object (Vemuri, 1988; Hecht-Nielsen, 1988). They have also been used to obtain a greater insight into the cognitive capabilities of the brain, such as searching, representation and learning. Ultimately, this understanding may be used to construct cognitive models which form the basis of artificial intelligence (Vemuri, 1988).

However, the areas in which neural networks are being used is expanding rapidly. In 1985, the world market for the commercial use of ANNs was virtually non-existent. At present, it is in the order of tens of millions of dollars and is expected to reach approximately $5 billion by the year 2000 (Hubick, 1992). In order to exploit the strengths of ANNs, the problems they are used for should have one or several of the following properties (Burke, 1991; Lippmann, 1987; Josin, 1987):

- A large amount of data is available.
- The input-generating distribution is unknown and probably non-Gaussian.

- It is expensive to estimate statistical parameters.
- Nonlinear relationships are suspected.
- The input data are noisy.
- A large number of attributes describe the inputs.
- On-line decision making is needed.
- Possible re-calibration of the system may be required.
- Uncertainty exists about the rules governing the mapping.
- Available models account for some, but not all of the data.
- Many hypotheses are pursued in parallel.
- High computation rates are required.
- Algorithms are unknown or intractable.
- Computations are time and resource intensive.

Neural networks provide a viable alternative to expert systems when the following conditions apply (Burke, 1991):

- Rules underlying decisions are not well understood.
- Numerous examples of the decisions are available.
- A large number of attributes describe the inputs.

Neural models may outperform statistical approaches when (Burke, 1991):

- Underlying distributions are unknown and assumptions of linearity or Gaussian data prove inadequate.
- A good deal of noise confounds analysis.
- Outliers may exist.

Different applications require the use of distinct classes of networks.

2.1.9.1 Networks for Pattern Recognition

When performing pattern recognition tasks, classification and association networks may be used. Classification networks determine which class best represents a set of input values. There are a number of networks that can be used for classification, all of which are based on Kohonen's method of self-organisation. Learning vector quantisation networks are the most popular of these (Lippmann, 1987).

Data association is very similar to classification. Association networks learn associations for error free or ideal patterns and then recognise noisy input patterns as

one of the learned patterns. Networks that can be used for this application include the bi-directional associative memory network, the Boltzmann machine, the Hamming network and the Hopfield network (NeuralWare, Inc., 1991).

In the field of pattern recognition, ANNs have been successfully used for:

- Recognising handwritten Chinese characters (Yeung and Fong, 1994).
- Alphabetical letter recognition (Omatu et al., 1990).
- Recognising music symbols (Ezhov and Sala, 1994).
- Recognising stock price patterns (Kamijo and Tanigawa, 1990).
- Target recognition (Mertzanis and Austin, 1991; Gilmore and Czuchry Jr., 1990).
- Contour recognition (Rangarajan et al., 1991; Loncelle, 1990; Oja et al., 1990; Thiaville et al., 1990; Lampinen and Oja, 1990).
- Texture recognition (Khotanzad and Lu, 1991; Gosh and Bovik, 1991; Rangarajan et al., 1991, Tirakis et al., 1990; Lampinen and Oja, 1990).
- Face recognition (Lampinen and Oja, 1990; Cottrell and Fleming, 1990).
- Classifying fingerprints (Mitra et al., 1994).
- Classifying two-dimensional shapes (Khotanzad and Lu, 1991).

2.1.9.2 Networks for Conceptualisation

Conceptualisation networks analyse input data in order to determine conceptual relationships. Adaptive resonance theory networks and self-organising maps can be used for this application (NeuralWare, Inc., 1991).

2.1.9.3 Networks for Filtering

Filtering encompasses the smoothing of an input signal. Recirculation networks can be used for this purpose (NeuralWare, Inc., 1991).

2.1.9.4 Networks for Optimisation

The use of neural networks for optimisation is discussed by a number of authors (NeuralWare, Inc., 1991; Burke and Ignizio, 1992; Krovvidy and Wee, 1990; Looi, 1992; Wang and Changkong, 1992; Vaithyanathan and Ignizio, 1992; Johnston and Adorf, 1992).

Artificial neural networks have the ability to represent and solve difficult combinatorial optimisation problems. Networks with non-adaptive learning rules are used for such

tasks. The most widely used of these networks is the Hopfield/Tank network (Hopfield and Tank, 1985).

In optimisation, energy functions resemble cost functions and have to be structured so that they encompass the objective function as well as the relevant constraints. The objective is to minimise this energy function in order to find the best solution to the original problem. To map a problem onto a neural network, the following steps need to be taken (Krovvidy and Wee, 1990):

1. A representation scheme has to be chosen which enables the decoding of the outputs into a solution to the problem.
2. An energy function has to be derived, the minimum value of which corresponds to the "best" solution to the problem.
3. Values have to be assigned to the parameters of the energy function.
4. Connectivities and input bias currents should be derived so that they appropriately represent the instance of the specific problem to be solved.

Neural network methods are capable of computing good, although not necessarily optimal, solutions. They provide viable alternatives to classical optimisation techniques for solving certain optimisation problems.

Reasons for using artificial neural network methods include (Looi, 1992; Wang and Changkong, 1992):

1. Their operation speed is high as a result of massively parallel computation.
2. They may be implemented by optical devices, which operate at higher speeds than conventional electronic circuits.
3. They have improved fault tolerance.
4. In contrast to methods based on the simplex algorithm, the solution procedure is not based on the sequential enumeration of extreme points.

As a result, the use of ANN methods is most appropriate for large-scale, linear programming problems (Wang and Changkong, 1992).

The major drawback with using ANN methods is that they tend to get stuck at local optima. One way to overcome this problem is to find many solutions, starting from different initial states, in an attempt to find the global optimum. However, this method is not very satisfactory and stochastic procedures, such as simulated annealing, can be used to "jolt" the network out of local minima.

Neural network methods are suited to solving resource constrained scheduling problems. In such problems the aim is to achieve a certain degree of scheduling effectiveness while satisfying the constraints imposed by one or more limited resources. Efficient scheduling is often of great importance in the business world. Recently, ANNs have been applied to a variety of scheduling tasks, including:

- Integrated scheduling of manufacturing systems (Dagli and Lammers, 1989).
- Planning and scheduling in aerospace projects (Ali, 1990).
- Space mission scheduling (Gaspin, 1989).
- Delivery truck scheduling (Daws et al, 1988).
- Large scale plant construction scheduling (Kobayashi and Nonaka, 1990).
- Scheduling of astronomical observations for the orbiting Hubble space telescope (Johnston and Adorf, 1992).

2.1.9.5 Networks for Prediction

The use of ANNs for prediction has been discussed by various authors (NeuralWare, Inc., 1991; Varfis and Versino, 1990; Jones et al., 1990; Windsor and Harker, 1990; Kimoto and Asakawa, 1990; Walter et al., 1990).

"Prediction" involves the estimation of output values given a set of input values. For example, evaporation rate can be predicted using temperature, humidity and wind velocity as inputs. Primarily, feed-forward networks with nonlinear transfer functions are used for prediction, of which back-propagation networks are the most common. Neural networks have been used successfully for complicated nonlinear forecasting tasks (Varfis and Versino, 1990) and time series prediction (Jones et al, 1990).

Hybrid methods combining regression techniques with neural network techniques can also be used with great success (Walter et al., 1990). Adaptive linear regression methods are used to calculate the predicted value at the next time step as a linear combination of previous values. However, in many cases the actual relation is nonlinear, so that a fixed set of linear prediction coefficients will not produce accurate results. To overcome this problem, a Kohonen network can be used to adaptively discretise the input data. If these discretisations are small enough, linear relations will be a good approximation in each cell, and a separate set of linear prediction coefficients can be estimated for each cell.

Examples of the use of artificial neural networks for prediction are given below.

A. General Applications

Economic time series prediction

Varfis and Versino (1990) used a multilayer feedforward network to predict manufacturing industries and energy production indices. Training was carried out using an on-line back-propagation algorithm. The training data consisted of daily manufacturing industries and energy production indices from January 1971 to August 1988. The results obtained were comparable with those obtained using ARMA type models.

Chakraborty et al. (1992) used a simple feed-forward network for the prediction of flour prices for the cities of Buffalo, Minneapolis and Kansas City. The neural network model performed better than a well known time series model described by Tiao and Tsay (1989).

Windsor and Harker (1990) used a standard back-propagation network for predicting the UK ordinary share index. The inputs used include the UK share index, interest rate, the M1 money supply, gross national product, personal disposable income, the balance of payments, the conservative party majority, the savings ratio, the birth rate, the unemployment rate, the US balance of payments and the Dow Jones index. Data from the previous 20 to 25 years was used for training, which was considered enough to capture all typical movement patterns. The back-propagation network was able to perform better than any of the alternative methods tried.

Kimoto and Asakawa (1990) used a back-propagation network with one hidden layer to learn the relationship between various technical and economical indices, and the timing for when to buy and sell stocks. The network was used to predict the best time to buy and sell for one month in advance. A standard sigmoid transfer function was employed. The indices that were used as inputs include the vector curve, the turnover, the interest rate, the foreign exchange rate and the New-York Dow Jones average. The neural network prediction system made an excellent profit in a simulation exercise, and obtained better results than a multiple regression analysis model.

Chaotic time series prediction

Lapedes and Farber (1988) used back-propagation networks to accurately predict future values of a chaotic time series, which frequently occur in nature. In this method,

past values of the series are used to predict future values, x(t+g), where g is some prediction time step in the future. As g increases, the prediction accuracy decreases. At large prediction time steps, g, neural network methods can be orders of magnitude more accurate than conventional methods (Lapedes and Farber, 1988).

The neural network determined the following input/output relationship:

$$o(t+g) = f(i_1(t), i_2(t-\Delta), \ldots, i_{N^I}(t-N^I\Delta)) \tag{2.36}$$

where $o(t+g)$ = the output at discrete time t+g
$i_1, \ldots, i_{N^I}$ = the network inputs
Δ = a time delay

The time series used by Lapedes and Farber was generated by integrating the Glass-Mackey equation shown below.

$$\dot{x} = \frac{ax(t-\tau)}{1 + x^{10}(t-\tau)} - b\,x(t) \tag{2.37}$$

where τ = the width of a time strip
a,b = constants

Training was carried out using 500 input/output training sets. Initially, values of $N^I = 4$ and $\Delta = 6$ were chosen, but it was found that various choices of N^I and Δ can affect the prediction accuracy by a factor of two.

Lapedes and Farber (1988) also determined that an iterative method is by far the most accurate method to use for large prediction steps. This involves predicting for a small value of g (say g/k, k = integer) initially and using the result of this prediction as the input to the next prediction step (i.e. at t + 2g/k). The iterative process is repeated until the predicted value at time t+g is obtained.

Predicting electrical power system loads

Atlas et al. (1990) used a multilayer perceptron network to successfully predict electrical power system loads from an hour to several days in the future. This is useful in the efficient scheduling and utilisation of power generation. The training set used consisted of 523 days of hourly temperature and load data from November 1, 1988 to January 27, 1989. The test set consisted of four days of hourly temperature and load

data from January 28 to January 31, 1989. The network inputs included the hour of the day, the two previous temperature readings and the two previous load readings. The output consisted of the current load. The predictions obtained were very accurate, with an average percentage error of 1.39%.

B. Applications in Water Engineering

The stochastic nature of many hydrological data sets, where a particular output is dependent on many unmeasurable actions, makes them suitable for analysis using artificial neural networks (Daniell, 1991). The following are potential application areas of artificial neural networks in the field of water engineering (Daniell,1991):

- Predicting flows, water consumption and evaporation.
- Filtering noisy data.
- Optimising water supply systems.
- Mapping radar rainfall images.
- Classifying data.
- Developing flood forecasting models.
- Developing flood warning systems.
- Developing streamflow models.
- Developing regional flood estimates.
- Evaluating water quality data.

ANNs are particularly suited to the prediction of water quantity and water quality parameters as:

- The prediction of one parameter is usually dependent on many other parameters.
- The exact relationship between parameters is generally unknown and is likely to be non-linear.
- Extensive data sets exist.
- The data is usually noisy as a result of measurement and sampling errors.

Examples where ANNs have been used in the field of water engineering include:

Estimation of water consumption / demand

Daniell (1991), compared the performance of a three-layer back-propagation network with that of an existing linear regression model used to estimate the monthly water consumption for Canberra, Australia. Network inputs included monthly rainfall, the

number of rain days per month, monthly evaporation and monthly average temperature. The output consisted of the average daily per capita water consumption for the month under consideration. Training data was taken from 1975 to 1984 and the test set consisted of data from 1985 and 1986. The network consisted of one output node, four input nodes and three hidden layer nodes. The results obtained using the ANN model compared favourably with those obtained using the linear regression model.

Fleming (1994) used an ANN model to forecast monthly water demands for the Northern Adelaide Plains, South Australia. The input variables that were found to be significant include the month of the year, the rainfall in the "current" month and the rainfall in previous months. The results obtained compared favourably with multiple regression and time series (Box-Jenkins) models.

Zhang et al. (1994) found that ANN models clearly outperformed multiple regression models, ARIMA models and Kalman filtering models for predicting daily water demands. The following five inputs were found to be dominant: daily maximum temperature, the weather condition (i.e. whether it was sunny, cloudy or rainy), precipitation, whether it was a weekday or a Sunday (including public holidays) and the last day's deliveries.

Canu et al. (1990) developed an ANN model to forecast daily water demand and compared the results obtained with those produced by stochastic and statistical methods. They found that ANNs were able to overcome the following limitations of statistical and stochastic methods:

- Stochastic models cannot accommodate inputs such as weather information or the day of the week.
- Both the stochastic and statistical models are linear in nature and hence cannot treat nonlinear phenomena such as steep consumption increases.

The ANN model used consisted of a fully-connected multi-layer perceptron network. The inputs used include the past seven consumption values, three pluviometer values, three temperature values and seven values representing the day of the week. The single output consisted of the predicted demand. The training set contained 12 years of data from 1976 to 1987, and data from 1988 were used for testing. The best value for the learning rate was determined by trial and error. The average prediction error was found to be very similar for all methods tested. However, the maximum prediction error is considered to be a better measure of the success of a forecasting tool. When considering this as the criterion, the ANN outperformed all other methods used. It should be noted, however, that the results obtained using the neural network method

were highly dependent on the learning rate. It was found that using small learning rates produced better results than using large learning rates.

Regional flood estimation

Daniell (1991) used a three-layer back-propagation network for regional flood estimation in the Australian Capital Territory, and compared the results with those obtained using multiple regression analysis. The ten catchment parameters that were used as inputs include area, slope, length, fall, precipitation, fractions of the catchment under urban, rural and forest use, annual series skew and partial series skew. There were seven nodes in the hidden layer and five nodes in the output layer representing the 2, 5, 10, 50 and 100 ARI flood flows. The hyperbolic tangent transfer function was used. The results obtained using the ANN model were very impressive and compared favourably with those obtained using a linear regression model.

Predicting salinity

DeSilets et al. (1992) developed a neural network model for predicting salinity in the Chesapeake Bay, U.S.A., and have compared its performance with that of a regression model. Predicting salinity values is important, as salinity affects the oyster populations in the area.

A total of ten neural network models were developed; five using bottom-of-the-bay-only data and five using data from all depths. Each of the five models corresponded to various regions of the bay; the upper bay, the middle bay, the lower bay, the tributaries and the entire bay. The regression model was used as a measure of comparison and to gain an understanding of the factors affecting salinity. It was found that location, depth and time of the year had a direct impact on salinity, and were used as inputs to the network. The solitary network output was the predicted salinity value. Training and testing data were collected from 34 stations throughout the bay, and a total number of 36,258 observations were available. The network used a back-propagation algorithm, a sigmoid transfer function, was fully connected and the number of nodes in the hidden layer varied from one to three. The optimum learning rate for the various models varied from 0.2 to 0.8, and the optimum momentum value varied from 0.1 to 0.9. These values were obtained by trial and error.

A stopping rule was used to determine the length of the training period. Training was stopped when one of the following occurred:

- The number of passes through the training set was at least 100 and the worst absolute error at the output node, once all patterns were considered, was less than 20%.
- The number of iterations reached 2,000.

Three main results were obtained:

1. The ANN model produced a lower mean absolute error than the regression model in 18 out of 20 cases.
2. The tails of the prediction error distribution produced by the neural network were shorter than those of the regression model. Hence the maximum errors obtained from the model were less.
3. The neural network model had more errors in the smallest error range and fewer errors in the largest error range.

Flow prediction

Lachtermacher and Fuller (1994) used ANN methods to obtain univariate flow predictions for the Gota River. The one-step ahead predictions were found to be slightly better than those obtained using other time series models, including ARMA models.

Karunanithi et al. (1994) developed a neural network model to predict the flow at an ungauged site in the Huron River, Michigan, using known flows at upstream and downstream locations. The predictions obtained using the neural network model were more accurate and less susceptible to noisy data than those obtained using a more conventional power model.

Runoff prediction

Zhu and Fujita (1994) successfully used ANN methods to obtain on-line and off-line runoff forecasts. The relationship between rainfall and runoff is dependent on many variables that interact with each other in a complex, usually non-linear, manner. Consequently, ANNs are well suited to this application.

Halff and Halff (1993) developed an ANN model for predicting runoff hydrographs given the rainfall at the 25 preceding time steps. The resolution used was five minutes. A three-layer network was used, with five nodes in the hidden layer and one node in the output layer. The data from four storms were used for training, while testing was

carried out using the data for one independent storm event. The results obtained were encouraging.

Prediction of nutrient concentrations

Daniell and Wundke (1993) used ANNs to successfully derive continuous records of total nitrogen and total phosphorous using continuous flow, conductivity, turbidity, temperature, oxidised nitrogen and soluble phosphorous records as inputs, in order to calculate nutrient loads of total nitrogen and total phosphorus. The neural network was found to perform better than a simple regression model.

Forecasting Minimum Temperature

Schizas et al. (1994) used a backpropagation network to forecast daily minimum temperatures at Larnaca Airport, Cyprus. The inputs used include the total cloud cover, wind direction, wind speed, visibility, the present weather conditions, atmospheric pressure, dry-bulb temperature, wet bulb temperature, the low cloud amount, astronomical day length and the observed minimum temperature of the previous night. The results obtained indicate that the neural network approach appears to be a viable alternative to traditional statistical techniques when limited historical data are available.

Design of a Reliable Groundwater Remediation Strategy

Ranjithan et al. (1993) used ANNs as a screening tool to assist in the design of reliable groundwater remediation strategies, which are determined with the aid of a management model. In many instances, the spatial variation of hydraulic conductivity is unknown, so that the management model has to be run for a large number of equally probable hydraulic conductivity distributions. In order to minimise the number of scenarios that have to be evaluated, a neural network model can be trained to predict the level of criticalness of a particular distribution of hydraulic conductivities, so that the management model only has to be run for the cases that are most critical.

2.2 Time Series Analysis

2.2.1 Introduction

A time series is "a collection of observations made sequentially in time." (Chatfield, 1975). The units of time used vary depending on the application (e.g. year, quarter, month, day, second). The observations may be continuous or discrete in time. Discrete time series may be obtained by sampling a continuous time series at regular intervals, or, if a variable does not have an instantaneous value, values may be aggregated over periods of time (Chatfield, 1975). The analysis of time series discussed in this review is concerned with discrete time series. Typically, the observations of a discrete time series made at time t are denoted by z_t.

The purpose of time series analysis is:

- To understand and evaluate the probability model or stochastic mechanism that gives rise to the time series under consideration (Cryer, 1986; Chatfield, 1975).
- To predict or forecast future values of the time series based on the history of the time series (Cryer, 1986).

2.2.2 Time Series Models

When developing a time series model, one is generally concerned with stochastic time series. In stochastic time series, "future values have a probability distribution which is conditioned by a knowledge of past values" (Chatfield, 1975). The basic steps in developing a time series model are given below (Cryer, 1986; Bowermann and O'Connell, 1979; Box and Jenkins, 1976):

1. Model identification:
In this step, the type and order of time series model that is appropriate for the given time series is tentatively identified. This is carried out by performing certain statistical analyses on the historical data, which enable a suitable model to be chosen. If more than one model appears suitable, the one with the fewest parameters should be chosen (parsimony).

2. Parameter estimation:
Once the tentative model has been identified, the unknown model parameters need to be estimated using techniques such as least squares estimation or maximum likelihood estimation.

3. Diagnostic checking:
In this step, the adequacy of the tentative model is assessed. This is performed by analysing how well the model fits the data and checking whether all assumptions have been reasonably satisfied. If the model proves to be inadequate, steps 1 and 2 above need to be repeated.

4. Forecasting:
Once it has been determined that a model is adequate, it can be used to forecast future values of the time series.

Criteria for a good model are discussed by a number of authors including Hendry and Richard (1983), Mizon and Richard (1986), Ericsson and Hendry (1985) and Harvey (1989). Some of these are given below:

1. Parsimony:
All other things being equal, the model with the fewest parameters is preferred.

2. Data coherence:
The model should be consistent with the data. This property is addressed in the diagnostic checking phase discussed above.

3. Consistency with prior knowledge:
The information derived from the model should be consistent with any prior knowledge about the time series.

4. Structural stability:
The model should provide a good fit to the data used in the parameter estimation phase as well as data that were not used to obtain the model parameters.

There are many different types of time series models. In this review, the focus will be on the ARMA (AutoRegressive - Moving Average) class of models. The vast majority of well developed, proven methods for implementing the model identification, parameter estimation and diagnostic checking stages are geared towards this class of models (Salas et al., 1985), and they have been widely used in many fields, including economics and water resources planning and management (Hipel, 1985). A brief discussion of more recent models is given in Section 2.2.13.10.

2.2.3 Linearity

Models of the ARMA type are usually linear. In a linear model, the output is a linear combination of the variables and white noise terms. This is in contrast to non-linear models, where at least one of the variable or white noise terms is raised to a power or is multiplied by another variable or white noise term (Hipel, 1985).

In general, non-linearities gain greater importance as the time interval between observations decreases (Hipel, 1985). A number of stochastic models have been developed to cater for non-linearities (e.g. Tong et al., 1985; Ozaki, 1985; Brillinger, 1985).

2.2.4 Distribution of Data

Generally, it is assumed that the data used to develop ARMA type models follow a Gaussian distribution (Irvine and Eberhardt, 1992). One way to test whether the data follow a Gaussian distribution is to use the probability plot correlation coefficient (PPCC). Filliben (1975) found that this compared favourably with seven other tests for normality. If the data follow a Gaussian distribution, the plot of the ordered observations versus the order statistic medians should be approximately linear. The PPCC test provides a measure of the linearity of the probability plot.

The PPCC test proposed by Filliben (1975) is only applicable to time series with less than 100 observations. However, Vogel (1986) extended the PPCC test for normality to sample lengths of 100 to 10,000.

When a time series is found to be non-Gaussian, it can either be transformed into a normally distributed time series, or a model can be fitted to the time series that takes its actual distribution into account (Hipel, 1985).

Data can be transformed to a Gaussian distribution using the Box-Cox transformation (Box and Cox, 1964). Conventional models are then fitted to the transformed series. Once modelling is complete, the time series has to be converted back into its original form. However, these backtransformations may introduce bias into the analysis, so that careful consideration should be given to transforming data which approach normality (Salas et al., 1980; Koch and Smillie, 1986). If it has been determined that transforming the data will improve model performance, various techniques (e.g. Matalas, 1967; Pankratz, 1983) can be used to minimise the bias (Rao et al., 1982).

Lewis (1985) describes a number of models that can be fitted to time series that follow the Exponential, Gamma, Weibull, Laplace, Beta or Mixed Exponential distributions, without first transforming them into normally distributed time series.

2.2.5 Spacing of Observations

When using the ARMA class of models, the observations have to be equally spaced in time. If this is not the case, appropriate techniques have to be used to produce a time series in which the observations are equally spaced (Hipel, 1985). Intervention analysis can be used if there are only a few missing observations (Baracos et al., 1981). If the gaps in the data are large, the seasonal adjustment approach suggested by McLeod et al. (1983) might be appropriate.

2.2.6 Stationarity

In order to fit a model to a particular time series, the series needs to be stationary. If a time series is stationary, the probability laws governing it do not change with time (i.e. the time series is in statistical equilibrium) (Cryer, 1986). Consequently for a stationary time series:

- The mean is constant with time (i.e. $\mu^{th}(t) = \mu^{th}$) or there is no trend.
- The variance is constant with time (i.e. $\sigma^2(t) = \sigma^2$).
- There are no seasonal or irregular fluctuations.

Trend is defined as the long term change in the mean (Chatfield, 1975). However, it is difficult to define what is meant by 'long term', as an apparent trend might be part of a cyclic variation with large period. Granger (1966) provides the following definition: "Trend in mean comprises all frequency components whose wavelength exceeds the length of the observed time series."

Seasonal variations are annual in period. Other cyclic variations might also have a fixed period (e.g. daily variation in temperature), or might be oscillations that do not have a fixed period but are predictable to some extent (Chatfield, 1975).

There are various methods for determining whether a time series is stationary or not, including plotting the data or using autocorrelation functions and partial autocorrelation functions. The various methods are discussed below.

2.2.6.1 Plotting the Data

The first step in any time series analysis is to plot the data. This gives a visual indication of whether the time series is stationary or not, as it is usually possible to detect seasonal effects, trends and changes in variance.

Apart from giving an insight into whether a time series is stationary or not, plotting the data enables the detection of outliers, discontinuities and turning points. If there are any outliers, they should be examined carefully to see whether there is justification for discarding them (Brockwell and Davis, 1987). If there are any discontinuities in the data, it might be advisable to split the time series into homogeneous segments and to model each segment separately (Brockwell and Davis, 1987).

2.2.6.2 The Autocorrelation Function

The autocorrelation function (ACF) can be used to determine whether a time series is stationary or not. The theoretical autocorrelation (ρ_k) "measures the relationship between any two time series observations separated by a lag of k time units" (Bowermann and O'Connell, 1979), and has the following properties:

- $-1 \leq \rho_k \leq 1$
- $\rho_k = \rho_{-k}$

Values of the theoretical autocorrelation near +1 or -1 indicate a strong linear dependence between the values considered, whereas a value close to 0 indicates very little correlation, which suggests that the values considered are independent.

The theoretical autocorrelation (ρ_k) is estimated using the sample autocorrelation (r_k). For the time series $z_t = z_a , z_{a+1} , z_{a+2} , \ldots\ldots\ldots, z_N$, the sample autocorrelation (r_k) is given by (Bowermann and O'Connell, 1979):

$$r_k = \frac{\sum_{t=a}^{N-k} (z_t - \bar{z})(z_{t+k} - \bar{z})}{\sum_{t=a}^{N} (z_t - \bar{z})^2} \qquad (2.38)$$

where $$\bar{z} = \frac{\sum_{t=a}^{N} z_t}{N-a+1} = \frac{\sum_{t=a}^{N} z_t}{n+1} \qquad (2.39)$$

The sample ACF (correlogram) is a plot of the sample autocorrelation against the lag k for $k > 0$. The maximum number of lags (K) for which the sample autocorrelation should be calculated is discussed by a number of authors. The various estimates for K are given below:

- $K \leq n/4$ (Chatfield, 1975; Hipel et al., 1977a).

- $K \leq 12$ for non-seasonal time series (Bowermann and O'Connell, 1979).

- $K \leq 4L$ for seasonal time series (Bowermann and O'Connell, 1979)
  where L = the number of seasons in a year
  (e.g. L = 12 for monthly data, L = 4 for quarterly data).

The correlogram is a very useful tool for examining dependence. Interpretation of the correlogram is vitally important and conveys a lot of information about the time series considered. However, the interpretation of correlograms can be difficult. Special features of correlograms for various types of time series are discussed below (Chatfield, 1975), which serves as a general guide for the interpretation of correlograms.

For a completely random series, one can expect values of r_k close to zero, provided a large number of observations are available. However, even if a time series is completely random, some values of r_k are generally significantly different from zero.

A stationary time series is considered to exhibit short term correlation if the correlogram consists of a reasonably large value of r_1 , followed by values of r_2, r_3 etc. that get progressively smaller. As the lag increases, r_k is approximately zero. If r_1, r_2 and r_3 are positive, larger than normal values of the series are more likely to be followed by other larger values, and similarly smaller values are more likely to be followed by smaller values.

When a series alternates, consecutive values are on different sides of the overall mean. The correlogram of such a series also alternates.

If a time series is non-stationary, the correlogram will die down extremely slowly (Bowermann and O'Connell, 1979). This is due to the fact that values on one side of the mean usually get followed by a large number of values on the same side of the mean as a result of the trend. If a time series exhibits seasonal or cyclic variation, it is also non-

stationary and the correlogram will have significant values of r_k (spikes) at the seasonal lags (Bowermann and O'Connell, 1979).

When a time series is stationary, the theoretical ACF tends to either:

- Die down with increasing lag k.
- Cut off after a particular lag $k = q^c$ (i.e. $\rho_k = 0$ for $k > q^c$).

Since ρ_k is approximated using r_k , it is difficult to know when $\rho_k = 0$, as r_k does not cut off completely. Guidelines to determine when $\rho_k = 0$ have been proposed by a number of authors.

Bowermann and O'Connell (1979) suggest that $\rho_k = 0$ if the following applies:

$$|r_k| \leq 2 \frac{1}{(N-a+1)^{1/2}} (1+2\sum_{j=1}^{q^c} r_j^2)^{1/2} \qquad \text{for } k > q^c \tag{2.40}$$

Whereas, Chatfield (1975) suggests that $\rho_k = 0$ if the following applies:

$$|r_k| \leq 2 \frac{1}{(N-a+1)^{1/2}} \qquad \text{for } k > q^c \tag{2.41}$$

2.2.6.3 The Partial Autocorrelation Function

The partial autocorrelation function (PACF) can also be used to determine whether a time series is stationary or not. The theoretical partial autocorrelation (ρ_{kk}) "may be thought of as the autocorrelation of any two observations, z_t and z_{t+k} , separated by a lag of k time units, with the effects of the intervening observations z_{t+1} , z_{t+2} , ...,z_{t+k-1} eliminated" (Bowermann and O'Connell, 1979). In other words, the theoretical partial autocorrelation (ρ_{kk}) is the correlation between the residuals of z_t and z_{t+k} after linear regression on z_{t+1} , z_{t+2} , ...,z_{t+k-1}.

The theoretical partial autocorrelation (ρ_{kk}) is estimated using the sample partial autocorrelation at lag k (r_{kk}). The sample partial autocorrelation is given by the following formula:

$$r_{kk} = \begin{cases} r_1 & \text{if } k=1 \\ \dfrac{r_k - \sum_{j=1}^{k-1} r_{k-1,j}\, r_{k-1}}{1 - \sum_{j=1}^{k-1} r_{k-1,j}\, r_j} & \text{if } k = 2, 3, \ldots \end{cases} \qquad (2.42)$$

where $r_{kj} = r_{k-1,j} - r_{kk}\, r_{k-1,k-j}$ for $j = 1,2,\ldots,k-1$

The sample PACF is a graph plotting r_{kk} for lags k=1,2,3,..... . The number of lags for which the sample partial autocorrelation should be plotted is the same as for the sample ACF. When a time series is stationary, the theoretical PACF tends to either:

- Die down with increasing lag k.
- Cut off after a particular lag $k = q^c$ (i.e. $\rho_{kk} = 0$ for $k > q^c$).

Since ρ_{kk} is approximated using r_{kk} , it is difficult to know when $\rho_{kk} = 0$, as r_{kk} does not cut off completely. Guidelines to determine when $\rho_{kk} = 0$ have been proposed by Bowermann and O'Connell (1979), who suggest that $\rho_{kk} = 0$ if the following applies:

$$|r_{kk}| \leq 2 \frac{1}{(N-a+1)^{1/2}} \qquad \text{for } k > q^c \qquad (2.43)$$

If the time series is non-stationary, the PACF will die down extremely slowly for non-seasonal time series and will exhibit 'spikes' at the seasonal lags for seasonal or cyclic time series.

In summary:

A. If a time series is stationary, the ACF and/or the PACF tend to either:

- Die down with increasing lag k.
- Cut off after a particular lag $k = q^c$ (i.e. $\rho_k = 0$ for $k > q^c$).

For non-seasonal time series, the following generally apply (Bowermann and O'Connell, 1979):

- If the ACF or the PACF does cut off, it does so when the lag $q^c \leq 2$.
- If the ACF or the PACF does not cut off when $q^c \leq 2$, then it generally dies down.

On the other hand, for a time series with seasonal or cyclic variation, the following applies (Bowermann and O'Connell, 1979):

- The time series may be considered stationary if the ACF or the PACF cut off after a lag $q^c \leq 2L+2$, where L is the number of seasons in a year (e.g. $L = 12$ for monthly data and $L = 4$ for quarterly data),but will frequently do so for $q^c \leq L+2$.

B. If a time series is non-stationary, the ACF and/or the PACF will:

- Die down extremely slowly if the time series under consideration does not possess seasonal or cyclic variation.
- Exhibit regular spikes at the seasonal lags if the time series under consideration possesses seasonal or cyclic variation.

If a time series is found to be non-stationary, it has to be transformed into a stationary time series before a tentative time series model can be chosen. A variety of transformations are available, some of which are discussed in Section 2.2.7.

2.2.7 Transformations

The two most common methods of transforming a non-stationary time series into a stationary time series include using the classical decomposition model and differencing.

2.2.7.1 The Classical Decomposition Model

The classical decomposition model (Chatfield, 1975; Brockwell and Davis, 1987) assumes that a time series consists of:

- A trend (mean function) component.
- A seasonal component.
- A random noise component.

The philosophy of the model is to first estimate and then eliminate the trend and seasonal components. The remaining random noise component, which is stationary, is then modelled. A purely random process with no time structure is called 'white noise', which is a sequence of independent, identically distributed random variables. For white noise in discrete time, the theoretical autocorrelation has the following properties:

$$\rho_k = \begin{cases} 1 & k = 0 \\ 0 & k = 1,2,3,.... \end{cases} \tag{2.44}$$

Using the classical decomposition model, time series commonly take on one of the following forms:

- $z_t = \bar{z}_t + s_t + e_t$ (2.45)

- $z_t = \bar{z}_t s_t + e_t$ (2.46)

- $z_t = \bar{z}_t s_t e_t$ (2.47)

where z_t = the observation at time t
$\bar{z}_t$ = the trend component
s_t = the seasonal component
e_t = the random noise component

In order to estimate and remove the trend and seasonal components, they, and the random noise component, have to be made additive, as is the case in Equation (2.45).

If the time series is completely multiplicative, as is the case in Equation (2.47), the variance increases with the mean. The variance can be stabilised and the components made additive by performing a logarithmic transformation:

$$u_t = \ln z_t \quad (2.48)$$

Time series where the seasonal effect is multiplicative but the random noise component is additive (Equation (2.46)) are difficult to transform. Ad hoc methods or the Box-Cox power transformation (Box and Cox, 1964) may be appropriate in this case.

Once the time series is in a form where the trend component, the seasonal component and the random noise component are completely additive, the first two components have to be estimated.

In the absence of seasonality, the trend component can be estimated using techniques such as curve fitting (Brockwell and Davis, 1987; Cryer, 1986; Chatfield, 1975) and filtering (Chatfield, 1975; Brockwell and Davis, 1987). If the time series possesses both trend and seasonality, the small trend method (Brockwell and Davis, 1987) or moving average estimation (Brockwell and Davis, 1987) may be used.

2.2.7.2 Differencing

Differencing is a widely used method for producing stationary time series, which involves taking the difference of data at appropriate time points (Box and Jenkins, 1976).

Elimination of trend in the absence of seasonality

The trend component can be eliminated by differencing. The following notation will be used (Brockwell and Davis, 1987):

- The first difference operator (∇) is given by:

$$\nabla z_t = z_t - z_{t-1} = (1-B)\, z_t \qquad (2.49)$$

where $B\, z_t = z_{t-1}$
B = the backshift operator

- The powers of B and ∇ are defined as follows (Brockwell and Davis, 1987):

$$B^d(z_t) = z_{t-d} \qquad (2.50)$$

$$\nabla^d = \nabla\,(\nabla^{d-1}\,(z_t)) \quad , d \geq 1 \qquad (2.51)$$

$$\nabla^0(z_t) = z_t \qquad (2.52)$$

- Polynomials involving ∇ and B are manipulated in exactly the same manner as polynomials involving real variables, for example:

$$\nabla^2 z_t = \nabla(\nabla z_t) = (1-B)(1-B)\, z_t = (1-2B+B^2)\, z_t = z_t - 2\, z_{t-1} + z_{t-2}$$

In general, non-stationary data are differenced until they become stationary. First or second order differencing is generally sufficient. In cases where the variance is multiplicative, it is necessary to perform a logarithmic transformation before differencing the data. However, care must be taken to avoid 'overdifferencing', which tends to introduce unnecessary correlations and hence may complicate a relatively simple model (Cryer, 1986).

Elimination of both trend and seasonality

In very simple cases, the seasonal component can be eliminated by differencing at the seasonal lag (Chatfield, 1975), e.g. for monthly data, the following transformation would be appropriate:

$$\nabla_{12} z_t = z_t - z_{t-12} \tag{2.53}$$

A more general model is described by Bowermann and O'Connell (1979). The following definitions apply:

- $\nabla^d z_t = (1-B)^d z_t$ (2.54)

  where d = the degree of non-seasonal differencing required to produce stationary time series values

- $\nabla_L^D z_t = (1-B^L)^D z_t$ (2.55)

  where L = the number of seasons in a year (e.g. L=12 for monthly data)
  D = any degree of seasonal differencing required to produce stationary time series values

Before the general transformation can be applied, the variance has to be constant or additive. If this is not the case, and the variance is multiplicative, the logarithmic transformation given in Equation (2.48) needs to be applied. Once the variance has been made additive, the following general transformation will produce a stationary time series, u_t:

$$u_t = \nabla_L^D \nabla^d z_t = (1-B^L)^D (1-B)^d z_t \tag{2.56}$$

Usually, only the following specific cases of the general transformation need to be considered:

- $d = 1, D = 0$: $u_t = \nabla_L^0 \nabla^1 z_t = z_t - z_{t-1}$ (2.57)
- $d = 1, D = 1$: $u_t = \nabla_L^1 \nabla^1 z_t = z_t - z_{t-1} - z_{t-L} + z_{t-1-L}$ (2.58)
- $d = 0, D = 1$: $u_t = \nabla_L^1 \nabla^0 z_t = z_t - z_{t-L}$ (2.59)

2.2.8 Model Identification

Once the time series has been made stationary, a choice has to be made about which probability model to use. Data familiarity plays a vital role in the model identification phase (Salas et al., 1980; Anderson, 1977). Generally, the ACF is used together with

the PACF to discern which model type is appropriate for the time series under consideration. This is the technique that will be discussed in this review.

Akaike's Information Criterion (AIC) (Akaike, 1973, 1974) is another method used for model identification. Other methods which are not discussed in this review include:

- The Criterion Autoregressive Transfer function (CAT) (Parzen, 1974).
- Extended autocorrelations (Tiao and Tsay, 1981).
- S and R arrays (Gray et al., 1978).
- Schwarz's approximation of the Bayes Information Criterion (BIC) (Schwarz, 1978).

Probability models for time series are collectively called stochastic processes, which are a collection of random variables ordered in time (Chatfield, 1975). Models that are in common use include:

- Moving Average (MA) models.
- AutoRegressive (AR) models.
- Mixed AutoRegressive-Moving Average (ARMA) models.
- Integrated Mixed AutoRegressive-Moving Average (ARIMA) models.
- General Multiplicative Seasonal Models.

The first step in deciding which of the above models is appropriate is to calculate the sample ACF and the sample PACF of the time series. The manner in which these functions die down (if they die down) and at what lag they cut off (if they cut off) can then be used to determine the type and order of model that is appropriate.

2.2.8.1 Non-Seasonal Models

Moving average model of order q

If $\{e_t\}$ is a random noise process with mean zero and variance σ^2 , then $\{z_t\}$ is a MA process of order q if:

$$z_t = e_t - \theta_1 e_{t-1} - \ldots\ldots\ldots - \theta_q e_{t-q} \qquad (2.60)$$

The process is called moving average because z_t is obtained by applying the weights (1, $-\theta_1$, $-\theta_2$,....., $-\theta_q$) to the variables (e_t , e_{t-1} , e_{t-2} ,....., e_{t-q}), and then moving the same weights one unit of time forward and applying them to e_{t+1} , e_t , e_{t-1} ,....., e_{t-q+1} to obtain z_{t+1} (Cryer, 1986).

A moving average process is characterised by:

- A PACF that dies down.
- An ACF that cuts off after lag q.

The manner in which the PACF dies down is typical for a particular type of model. Examples are given by Bowermann and O'Connell (1979) and Box and Jenkins (1976).

The behaviour of the ACF and the PACF are used to determine which model order is appropriate. If it is difficult to determine the model order using the ACF and the PACF, the following procedure may be used (Chatfield, 1975):

1. Fit processes of successively higher order to the data.
2. Calculate the sum of squares of the residuals of each order.
3. Plot the sum of squares against the order.

As the number of parameters is increased, the residual sum of squares always decreases. However, by plotting the sum of squares of the residuals against the order, the point where additional parameters result in very little improvement can be identified. This process requires a significant amount of computation and is often unnecessary.

Autoregressive model of order p

If $\{e_t\}$ is a purely random process with mean zero and variance σ^2 , then $\{z_t\}$ is an AR process of order p if:

$$z_t = \phi_1 z_{t-1} + \phi_2 z_{t-2} + \ldots\ldots + \phi_p z_{t-p} + e_t \qquad (2.61)$$

The current value that is given by the model is a linear combination of the p past values of the time series plus an 'innovation' term (e_t), which accounts for everything that is not explained by past valucs. This is very similar to multiple regression analysis, but z_t is not regressed on independent variables, but on past values of z_t, hence the name autoregressive.

Any AR process of finite order can be expressed as a MA process of infinite order. For example, the first order AR process may be expressed as an infinite order MA process of the form:

$$z_t = e_t + \alpha e_{t-1} + \alpha^2 e_{t-2} + \ldots\ldots \quad , \text{(provided } -1<\alpha<+1) \qquad (2.62)$$

An autoregressive process is characterised by:

- A PACF that cuts off after lag p.
- An ACF that dies down.

The manner in which the ACF dies down is typical for a particular type of model. Examples are given by Bowermann and O'Connell (1979) and Box and Jenkins (1976).

The behaviour of the ACF and the PACF are used to determine which model order is appropriate. Significant values of the PACF will generally indicate the order of the process, as the PACF of a pth order AR process is zero for all lags greater than p. If it is difficult to determine the order using the ACF and the PACF, the sum of squares of the residuals may be computed for processes of increasing order, as described above.

Mixed autoregressive - moving average model of order (p, q)

Mixed autoregressive - moving average (ARMA) models of order (p,q) consist of AR terms of order p and MA terms of order q. Although most time series can be adequately described by an AR model or a MA model, it usually requires fewer parameters to describe the same time series using a mixed model. A mixed ARMA model of order (p,q) is defined as:

$$z_t = \phi_1 z_{t-1} + \phi_2 z_{t-2} + \ldots\ldots + \phi_p z_{t-p} - \theta_1 e_{t-1} - \ldots\ldots\ldots - \theta_q e_{t-q} + e_t \qquad (2.63)$$

An mixed ARMA process is characterised by:

- A PACF that dies down.
- An ACF that dies down.

The manner in which the ACF and the PACF die down is typical for a particular type of model. Examples are given by Bowermann and O'Connell (1979) and Box and Jenkins (1976).

Integrated mixed autoregressive - moving average (ARIMA) model

"A series $\{z_t\}$ is said to follow an ARIMA model if the dth difference $u_t = \nabla^d z_t$ is a stationary ARMA process." (Cryer, 1986). The ARIMA process is of the form:

$$u_t = \phi_1 u_{t-1} + \phi_2 u_{t-2} + \ldots\ldots + \phi_p u_{t-p} - \theta_1 e_{t-1} - \ldots\ldots\ldots - \theta_q e_{t-q} + e_t \qquad (2.64)$$

The name integrated model derives from the fact that the stationary model which is fitted to the differenced data had to be summed, or 'integrated', to provide a model for the non-stationary data (Chatfield, 1975).

If u_t is described by an ARMA process of order (p,q), then z_t is described by an ARIMA process of order (p,d,q). Usually d = 1 or d = 2 is sufficient.

2.2.8.2 Seasonal Models

There are two types of models that deal with data with seasonal variation; namely Periodic AutoRegressive - Moving Average (PARMA) models and general multiplicative seasonal models.

PARMA model

PARMA models are extensions to ARMA models that allow different models to be fitted for different seasons (Vecchia, 1985). The term season is defined as any time resolution smaller than one year. Essentially PARMA models consist of a series of ARMA models; using a different model for each season (Hipel, 1985).

PARMA models may be represented as follows (Vecchia, 1985):

$$z_{n'L+\nu} = \phi_1(\nu)\, z_{n'L+\nu-1} + \phi_2(\nu)\, z_{n'L+\nu-2} + \ldots\ldots + \phi_{p(\nu)}(\nu)\, z_{n'L+\nu-p(\nu)} - \theta_1(\nu)\, e_{n'L+\nu-1} - \theta_2(\nu)\, e_{n'L+\nu-2} \ldots\ldots\ldots - \theta_{q(\nu)}(\nu)\, e_{n'L+\nu-q(\nu)} + e_{n'L+\nu} \qquad (2.65)$$

where

$\{z_{n'L+\nu}\}$	=	a seasonal time series
L	=	the number of seasons per year
n'	=	the year index (n = 0, ±1, ...)
ν	=	the season index (ν = 1, 2, ..., S)
$\{e_{n'L+\nu}\}$	=	independent, zero mean variables with Var $\{e_{n'L+\nu}\} = \sigma^2(\nu) > 0$
$\phi_1(\nu), \phi_2(\nu), \ldots$	=	the seasonal AR parameters
$\theta_1(\nu), \theta_2(\nu), \ldots$	=	the seasonal MA parameters
$\sigma^2(\nu)$	=	the seasonal variance

It should be noted that:

1. The AR orders (p(v)) and the MA orders (q(v)) are also allowed to vary with the seasons.

2. It is assumed that $\{z_{n'L+v}\}$ has finite second order moments and that $E(z_{n'L+v}) = 0$ for all n' and v. If the latter assumption does not hold, $z_{n'L+v}$ should be replaced by $z_{n'L+v} - E(z_{n'L+v})$.

PARMA models will not be discussed further in this review, as systematic procedures for determining the order of the AR and MA components are not well developed. A detailed description of the PARMA modelling procedure is given by Vecchia (1985), Tao and Delleur (1976) and Salas et al. (1982).

General multiplicative seasonal model

With this type of model, the data is first deseasonalised, and an appropriate non-seasonal stochastic model is then fitted to the deseasonalised data (Thompstone et al., 1985). Multiplicative seasonal models have been frequently used for modelling seasonal data (Irvine and Eberhardt, 1992; Rao et al., 1982; Vandaele, 1983).

In order to identify the appropriate multiplicative seasonal model, the following notation needs to be introduced (Bowermann and O'Connell, 1979):

- By including the backshift operator (B), the mixed ARMA model of order (p,q) (Equation 2.63) can be expressed as:

$$(1 - \phi_1 B - \phi_2 B^2 - ..... - \phi_p B^p)\, z_t = (1 - \theta_1 B - \theta_2 B^2 - ..... - \theta_q B^q)\, e_t \qquad (2.66)$$

- The non-seasonal AR operator of order p is defined as:

$$\phi_p(B) = (1 - \phi_1 B - \phi_2 B^2 - ..... - \phi_p B^p) \qquad (2.67)$$

- The non-seasonal MA operator of order q is defined as:

$$\theta_q(B) = (1 - \theta_1 B - \theta_2 B^2 - ..... - \theta_q B^q) \qquad (2.68)$$

- The non-seasonal mixed ARMA model of order (p,q) can therefore be written as:

$$\phi_p(B)\, z_t = \theta_q(B)\, e_t \qquad (2.69)$$

- The seasonal AR operator of order P is given by:

$$\phi_P(B^L) = (1 - \phi_{1,L} B^L - \phi_{2,L} B^{2L} - ..... - \phi_{p,L} B^{PL}) \qquad (2.70)$$

- The seasonal MA operator of order Q is given by:

$$\theta_Q(B^L) = (1 - \theta_{1,L}B^L - \theta_{2,L}B^{2L} - ..... - \theta_{Q,L}B^{QL}) \quad (2.71)$$

Using the terms defined above, the general multiplicative seasonal model, which is a mixed ARMA model of order (p, P, q, Q), can be expressed as:

$$\phi_p(B)\ \phi_P(B^L)\ z_t = \theta_q(B)\ \theta_Q(B^L)\ e_t \quad (2.72)$$

In order to determine which form of the general multiplicative seasonal model most adequately describes the time series under consideration, one needs to determine:

- Which of the operators $\phi_p(B)$, $\phi_P(B^L)$, $\theta_q(B)$ and $\theta_Q(B^L)$ are appropriate.
- The order of each operator that is included.

As with non-seasonal models, model identification is carried out with the aid of the ACF and the PACF. The following two methods can be employed.

Method 1

This is similar to the method used for non-seasonal models. The ACF and the PACF can be compared with those given by the various special forms of the general multiplicative seasonal model. A summary of these is given in Box and Jenkins (1976). However, in practice this comparison is difficult to carry out.

Method 2

This method is discussed by Bowermann and O'Connell (1979) and is somewhat intuitive. Several iterations of the procedure may be required to arrive at a satisfactory model. The general procedure involves the following steps:

1. The ACF and the PACF are calculated.
2. The behaviour of the ACF and the PACF are used to identify the most obvious component of the tentative model in accordance with the guidelines given below.
3. The adequacy of the model is checked and if the proposed model is inadequate, the next most obvious component has to be chosen and added to the model.

Some guidelines for choosing appropriate components are given below (Bowermann and O'Connell, 1979):

- If - the sample autocorrelation dies down fairly quickly and
  - the sample PACF cuts off after a lag that is substantially less than L (perhaps ≤ 2).

  Use - some form of the non-seasonal AR operator $\phi_p(B)$. The order, p, that should be used is determined from the lag at which the sample partial autocorrelation function cuts off (e.g. if it cuts off after lag q^c, $p = q^c$).

- If - the sample autocorrelation dies down fairly quickly and
  - the sample PACF cuts off after a lag that is nearly equal to L (e.g. L, L+1, 2L, 2L+1, 2L+2) and
  - no spikes exist in the PACF at lags substantially smaller than L (perhaps at lags 1 and 2).

  Use - some form of the seasonal AR operator $\phi_P(B^L)$. The order p is determined by the lags at which spikes occur in the sample ACF.

- If - the sample autocorrelation dies down fairly quickly and
  - the sample PACF cuts off after a lag that is nearly equal to L (e.g. L, L+1, 2L, 2L+1, 2L+2) and
  - spikes exist in the PACF at lags substantially smaller than L (perhaps at lags 1 and 2).

  Use - some form of the product of the non-seasonal AR operator and the seasonal AR operator $\phi_p(B)\ \phi_P(B^L)$. The order is determined by the lags at which the spikes occur in the sample PACF.

- If - the sample autocorrelation dies down fairly quickly and
  - the sample ACF cuts off after a lag that is substantially less than L (e.g. ≤ 2).

  Use - some form of the non-seasonal MA operator $\theta_q(B)$. The order q that should be used is determined from the lag at which the sample ACF cuts off.

- If - the sample PACF dies down fairly quickly and
  - the sample ACF cuts off after a lag that is nearly equal to L (e.g. L, L+1, 2L, 2L+1, 2L+2) and

- no spikes exist in the ACF at lags substantially smaller than L (perhaps at lags 1 and 2).

Use - some form of the seasonal MA operator $\theta_Q(B^L)$. The order, Q, is determined by the lags at which spikes occur in the sample ACF.

- If - the sample PACF dies down fairly quickly and
  - the sample ACF cuts off after a lag that is nearly equal to L (e.g. L, L+1, 2L, 2L+1, 2L+2) and
  - spikes exist in the ACF at lags substantially smaller than L (perhaps at lags 1 and 2).

Use - some form of the product of the non-seasonal MA operator and the seasonal MA operator $\theta_q(B)\,\theta_Q(B^L)$. The order is determined by the lags at which the spikes occur in the PACF.

- If - the sample autocorrelation dies down fairly quickly and
  - the sample partial autocorrelation dies down fairly quickly.

Use - some form of both operators $\theta_q(B)\,\theta_Q(B^L)$ and $\phi_p(B)\,\phi_P(B^L)$. Usually a simple form is sufficient.

2.2.8.3 Akaike's Information Criterion (AIC)

Up to this point, the use of the ACF and the PACF has been discussed as a means for model identification. Akaike's information criterion (Akaike, 1974) is an alternative method for identifying an appropriate tentative model. Akaike's information criterion is given by:

$$\text{AIC} = -2 \log (\text{maximum likelihood}) + 2n_p \qquad (2.73)$$

where n_p = the total number of AR and MA parameters

Akaike's information criterion is evaluated for a range of possible models, and the model chosen is the one with the lowest value of AIC (Thompstone et al., 1985). The term $2n_p$ in Equation (2.73) serves as a penalty function to avoid consideration of models that have too many parameters. The advantage of using the AIC is that it takes into account two of the most important modelling principles, namely parsimony and the goodness of the statistical fit (Hipel, 1985).

2.2.9 Parameter Estimation

Once the type and order of the tentative model that best describes the time series under consideration have been determined, the unknown model parameters have to be estimated. Different methods can be used to estimate the parameters (Cryer, 1986) including:

- The method of moments.
- Least-squares estimation.
- Maximum likelihood estimation.
- Unconditional least-squares estimation.

The model parameters (ϕ_p, ϕ_P, θ_q and θ_Q) and the theoretical ACF are related, so that the parameters can be expressed in terms of ρ_k. The particular relationship depends on the type of model that is being used.

Preliminary estimates of the parameters ϕ_p, ϕ_P, θ_q and θ_Q are made by using the sample autocorrelation (r_k) instead of the theoretical autocorrelation (ρ_k).

Before the preliminary estimates of the parameters can be used to produce the final estimates, they have to meet certain stationarity and invertibility conditions. This ensures that the particular model obtained is unique for a given ACF. Stationarity and invertibility conditions for a variety of models are discussed by Chatfield (1975). The preliminary parameter estimate that satisfies the imposed stationarity and invertibility conditions is then used as an input to a computer program, which produces the final parameter estimates (Bowermann and O'Connell, 1979). The computer applies an iterative procedure to obtain the final estimates in accordance with the least-squares error criterion, which gives parameters such that the squares of the errors between the actual and predicted values are minimised. Standard computer packages are widely available which perform least-squares estimation.

It should be noted that it is sometimes difficult to obtain preliminary parameter estimates in terms of the sample autocorrelation. This is especially the case for seasonal models. In those situations a preliminary estimate of 0.1 may be used for all parameters (Bowermann and O'Connell, 1979). Using preliminary estimates of 0.1 is acceptable as:

- Preliminary estimates of 0.1 satisfy all stationarity and invertibility conditions.
- The final estimates produced using least-squares analysis are not sensitive to the preliminary inputs.

2.2.10 Diagnostic Checking

Once a particular model has been tentatively chosen and the unknown model parameters have been estimated, the adequacy of the proposed model has to be evaluated. This is done by examining the errors produced by the model (Bowermann and O'Connell, 1979), which are caused by:

- The irregular component.
- The inability to perfectly predict the trend, seasonal and cyclic components of the time series.

The error is more commonly called the residual, which is the difference between the actual and the fitted values. If the model adequately describes the data, the residuals should be almost uncorrelated. However, the residuals $\{R_t\}$ tend to have different properties to true white noise.

For a large sample, the sample autocorrelation of true white noise is approximately uncorrelated, and the data are normally distributed with mean zero and variance 1/n.

The residuals are also normally distributed with mean zero. However, for small lags, the variance can be substantially less than 1/n and the estimates of r for the various lags can be highly correlated. For larger lags, the variance approximates 1/n and the estimates of r are uncorrelated (Cryer, 1986).

The first step in any residual analysis is to inspect the graph of residuals against time (Cryer, 1986). If the model is adequate, the plot should consist of a rectangular scatter of values about a zero mean level without trends.

The second step is to check the normality of the residuals by plotting a histogram of residual distribution about the mean (Cryer, 1986; Hipel, 1985). The variance of the residuals should also be constant (homoscedasticity). If the residuals are non-normal or do not have constant variance, the original data should be transformed using a Box-Cox transform (Hipel, 1985).

Various additional tests have been developed to determine whether the model obtained adequately describes the time series. Some of these tests are described below.

2.2.10.1 The Durbin - Watson Statistic

This test examines the residuals. The Durbin - Watson statistic is a test for first order correlation only and is given by (Chatfield, 1975):

$$d^{DW} = \frac{\sum_{t=2}^{n}(z_t - z_{t-1})^2}{\sum_{t=1}^{n} z_t^2} \tag{2.74}$$

The Durbin - Watson statistic is related to the first autocorrelation coefficient of the residuals in the following manner:

$$d^{DW} \approx 2\,(1 - r_1) \tag{2.75}$$

If d^{DW} exceeds a critical value, the model is deemed to be inadequate. Critical values of d^{DW} depend on the number of independent variables in the model (Kendall, 1973). Tables of these values are available

Similar statistics can also be calculated for higher order correlations. The following statistic can be used for fourth order correlations:

$$d_4^{DW} = \frac{\sum_{t=5}^{n}(z_t - z_{t-4})^2}{\sum_{t=1}^{n} z_t^2} \tag{2.76}$$

This value is related to the fourth autocorrelation coefficient in the following manner:

$$d_4^{DW} \approx 2\,(1 - r_4) \tag{2.77}$$

2.2.10.2 The Box - Pierce Method

This method was devised by Box and Pierce (1970) and involves the calculation of the correlogram for the residuals. The correlogram is then checked to see if there are any significant correlations, in which case the proposed model is inadequate. Values of r_k are deemed to be significant if they lie outside the range $\pm 2/\sqrt{n}$. For low lags, values

of r_k inside the range $\pm 2/\sqrt{n}$ may still be significant and may require further investigation.

2.2.10.3 The Portmanteau Lack-of-fit Test

Unlike the previous methods, in which the autocorrelations of the residuals were examined separately, this method investigates the first k' values of r_k at the same time. This is achieved by using the Box-Pierce chi-square statistic (Q^{BP}), which is given by (Bowermann and O'Connell, 1979):

$$Q^{BP} = (n - d') \sum_{i=1}^{k'} r_i^2 \qquad (2.78)$$

where n = the number of observations in the original time series
d' = the number of observations 'lost' as part of differencing
r_i = the sample autocorrelation of the residuals at lag i

If the residuals are highly correlated, Q^{BP} will be large, and the proposed model is unacceptable. The model chosen is appropriate if:

$$Q^{BP} < \chi^2_{.5}(k' - n_p) \qquad (2.79)$$

where n_p = the number of parameters that must be estimated in the model
k' = the number of residual autocorrelations used in the calculation of Q^{BP}
$\chi^2_{.5}(k' - n_p)$ = the point on the chi-square distribution having $(k' - n_p)$ degrees of freedom such that there is an area of 0.05 under the curve of this distribution above this point

The value of k' is chosen arbitrarily. However, Bowermann and O'Connell (1979) give the following guidelines:

- For non-seasonal models: k' = 12 (possibly 24 and 36).
- For seasonal models: k' = 3L.

The modified Box - Pierce statistic (or Ljung - Box - Pierce statistic) was proposed by Ljung and Box (1978) and is given by:

$$Q^{BP*} = n(n+2)\sum_{i=1}^{k'}\frac{r_i^2}{n-i} \qquad (2.80)$$

This statistic usually gives better results (Cryer, 1986).

2.2.11 Forecasting

Forecasts of future values can be made once the type and order of the model have been chosen, the model parameters have been estimated and the adequacy of the model has been checked. When forecasting, the pattern that has been found to adequately describe the historical data in the model development process is extrapolated to prepare forecasts of future values. The assumption is that the pattern that has been identified will continue in the future.

For a given time series $z_t = z_a , z_{a+1} , z_{a+2} , ...,z_N$, the aim of forecasting is to predict the value z_{n+g} , where g is called the forecasting period. The longer the forecasting period, the greater the chance that the underlying model might change and hence the more unreliable the forecast.

If a forecast $z_{o'+g}$ is made from time origin o', the following applies:

- If $g \leq 0$, the error term $e_{o'+g}$ can be estimated.
- If $g > 0$, the error term $e_{o'+g}$ cannot be estimated and is set equal to 0.

2.2.12 The Box-Jenkins Methodology

The Box-Jenkins methodology (Box and Jenkins, 1976) provides a framework for fitting a univariate ARIMA model to a given time series. The methodology incorporates many of the concepts discussed in this chapter and is summarised in Figure 2.17. The following points should be noted:

- The sample ACF and the sample PACF are used to determine whether the time series is stationary as described in Sections 2.2.6.2 and 2.2.6.3.

- Non-stationary time series are transformed into stationary time series by means of differencing (Section 2.2.7.2).

- If the data are non-seasonal, an ARMA model is fitted to the differenced time series, u_t (Section 2.2.8.1).

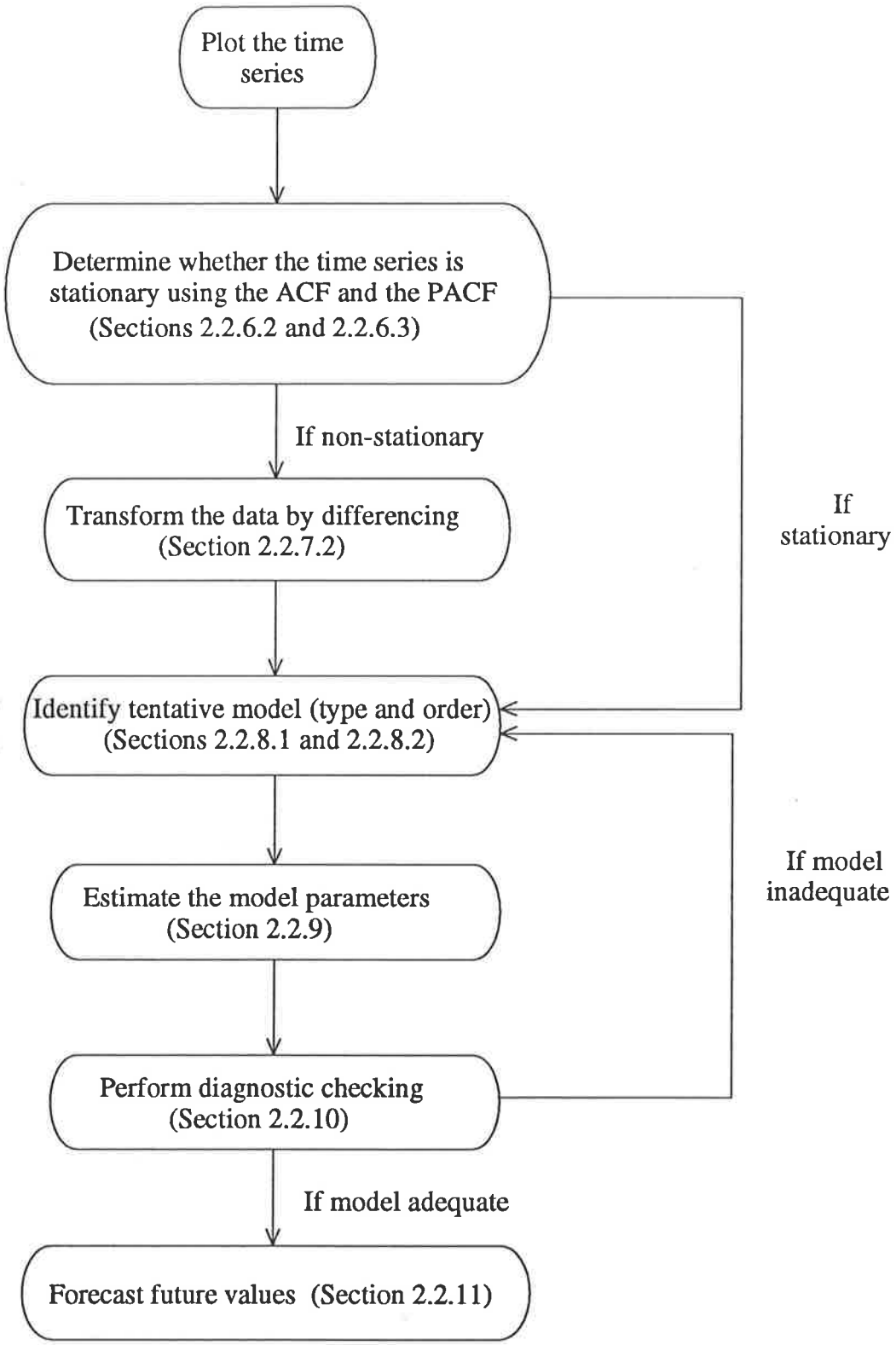


Figure 2.17: Summary of the Box-Jenkins Methodology

- If the data are seasonal, a general multiplicative seasonal model is fitted to u_t (Section 2.2.8.2). It should be remembered that the order of the model (p,P,q,Q) is determined with the aid of the sample ACF and the sample PACF. Values for p and P are chosen by looking at the first few values of the sample ACF and the sample PACF, whereas q and Q are chosen by looking at values of the sample ACF and the PACF function at seasonal lags.

- The model parameters for the particular model chosen are estimated using a method such as least squares estimation (Section 2.2.9).

- Diagnostic checking is carried out using the Portmanteau lack-of-fit test (Section 2.2.10.3).

Advantages of the Box-Jenkins methodology include the following (Chatfield, 1975):

- There are a wide number of model classes available.
- It produces more accurate forecasts than other methods.

Disadvantages of the Box-Jenkins method include the following (Chatfield, 1975):

- It is fairly complicated.
- It requires considerable skill and experience.
- It requires lengthy computer programs and time.
- It is expensive.

2.2.13 Multivariate Time Series Models

2.2.13.1 Introduction

Multivariate time series models deal with several time series that are interrelated. Frequently, variables not only depend on their own past values, but also depend on past values of other variables.

When analysing multivariate time series, the aim is to determine which variables are interrelated and at what lags, resulting in a model describing the relationship between the variables. Multivariate time series are utilised where *a priori* knowledge is available suggesting that the various time series are linked together.

Examples of the interrelation between several variables are found in a variety of fields such as economics and engineering. An example in the field of economics includes the prediction of the share market index, which depends on economic variables such as

interest rate, foreign exchange rate, gross national product, balance of payments, personal disposable income, investment and unemployment rate (Windsor and Harker, 1990; Kimoto and Asakawa, 1990; Harvey, 1989).

Examples in the field of engineering include the prediction of water consumption, floods and salinity. Water consumption depends on monthly rainfall, the number of rain days per month, monthly evaporation and monthly average temperature (Daniell, 1991). Regional floods depend on various catchment parameters such as area, slope, length, fall, precipitation, the fraction of the catchment under urban, rural and forest use, annual series skew and partial series skew (Daniell, 1991). River salinity at a particular location depends on various upstream salinities, river levels and flows (Maier and Dandy, 1993).

As is the case with univariate time series models, multivariate time series models are only fitted to variables which are stationary (Harvey, 1981). In fact, most of the basic theory applying to univariate time series models extends to multivariate time series models (Brockwell and Davis, 1987). The basic steps of model identification, parameter estimation, diagnostic checking and forecasting used in developing univariate time series models still apply to multivariate time series models. However, some new problems arise in the multivariate case. In particular, the model identification process is not as well defined and requires a trial and error approach combined with a great deal of judgement.

In general, one would expect multivariate models to outperform univariate models, as univariate models simply extrapolate past events. However, univariate models often provide a valuable yardstick against which the performance of multivariate models can be assessed (Harvey, 1989).

2.2.13.2 Classes of Multivariate Time Series Models

There are a number of classes of multivariate time series models of the ARMA type (Salas et al., 1985), including:

1. Vector autoregressive - moving average (VARMA) models.
2. Contemporaneous autoregressive - moving average (CARMA) models.
3. Transfer function autoregressive - moving average (TFN) models.
4. Exogenous-variable autoregressive - moving average (ARMAX) models.
5. Periodic autoregressive - moving average (PARMA) models.

VARMA models

The general VARMA (p,q) process for N' time series is described by the following equation (Harvey, 1981):

$$\mathbf{z}_t = \Phi_1\mathbf{z}_{t-1} + \ldots\ldots + \Phi_p\mathbf{z}_{t-p} + \mathbf{e}_t - \Theta_1\mathbf{e}_{t-1} - \ldots\ldots - \Theta_q\mathbf{e}_{t-q} \qquad (2.81)$$

where
$\mathbf{z}_t$ = a N' x 1 vector of observations
$\mathbf{e}_t$ = a N' x 1 vector of noise elements
Φ_i = a N' x N' matrix of AR parameters
Θ_j = a N' x N' matrix of MA parameters

A VARMA model fitted to a set of differenced time series is known as a Vector AutoRegressive Integrated Moving Average (VARIMA) model.

When developing VARMA models, the parameter matrices consist of non-zero elements (Ledolter, 1978). This indicates that feedback relationships exist, in the sense that if one time series causes another, the reverse is also true. An example of such a relationship can be found between time series of precipitation, lake evaporation and lake storage for some large lakes (Salas et al., 1985).

However, there are a number of problems associated with models of the VARMA type (Salas et al., 1985; Camacho et al., 1985, 1983), including:

- They generally contain a large number of parameters, making it difficult to estimate them efficiently, especially when limited data sets are available.
- The relationships found in many physical systems, including those in the fields of hydrology and water resources, make the inclusion of feedback parameters unjustified.

In order to overcome the above problems, CARMA and TFN models, which are special cases of VARMA models, have been introduced.

CARMA models

CARMA models result when all but the diagonal elements in the parameter matrices of a VARMA model are zero (Salas et al., 1985). CARMA models are essentially a collection of N' univariate ARMA models with contemporaneously correlated error terms (Camacho et al., 1985). When developing CARMA models, univariate

techniques are used to fit ARMA models to each time series. A multivariate model is then fitted to the residuals of the univariate models (Salas et al., 1985).

A property of CARMA models is that only concurrent, or contemporaneous variables, of the different time series are correlated, which is referred to as instantaneous causality (Granger, 1969). The annual or seasonal precipitation time series at several locations in a particular region may be contemporaneously correlated (Salas et al., 1985).

TFN models

TFN models result when the parameter matrices in a VARMA model are upper or lower triangular (Salas et al., 1985; Camacho et al., 1985). This indicates unidirectional (one-way) causality, which means that one time series causes another, but the reverse is not true.

An example of such a relationship is that of three streamflow time series at different locations along a river. The flow at the site located furthest upstream causes the flow at both downstream sites. The flow at the intermediate site is caused by the flow at the upstream site and causes the flow at the downstream site. The flow at site located furthest downstream is caused by the two upstream flows but has no effect on either of them. Each variable is also assumed to be affected by its own past values (Salas et al., 1985).

ARMAX models

ARMAX models (Cooper and Wood, 1982a, 1982b) are an extension to ARMA models that take exogenous variables into account. ARMAX models are multiple input-output models and are described by the following equation (Salas et al., 1985):

$$\begin{aligned} z_t &= \Phi_1 z_{t-1} + \ldots + \Phi_p z_{t-p} + e_t - \Theta_1 e_{t-1} - \ldots - \Theta_q e_{t-q} + \\ &\quad \Gamma_1 v_{t-1} + \ldots + \Gamma_r v_{t-r'} \end{aligned} \tag{2.82}$$

where
- z_t = a N' x 1 vector of observations
- e_t = a N' x 1 vector of noise elements
- v_t = a K' x 1 vector of exogenous variables
- Φ_i = a N' x N' matrix of AR parameters
- Θ_j = a N' x N' matrix of MA parameters
- Γ_k = a N' x K' matrix of parameters for the exogenous variables

PARMA models

A PARMA model (Section 2.2.8.2) may be classed as a special kind of multivariate model, as it can be expressed as a VARMA model and is made up of a number of models; one for each season (Salas et al., 1985; Vecchia, 1985).

2.2.13.3 The Cross-Correlation Function

The cross-correlation function (CCF) can be used to determine:

- Whether the time series under consideration are jointly stationary.
- The strength of the relationship between two time series, and at what lag(s) maximum correspondence occurs (Brockwell and Davis, 1987).

The CCF is simply a plot of the cross-correlation between two time series against the lag (k).

The following steps are required to calculate the CCF between two time series $z_{i,t} = z_{i,a}, z_{i,a+1}, z_{i,a+2}, \ldots\ldots, z_{i,N}$ and $z_{j,t} = z_{j,a}, z_{j,a+1}, z_{j,a+2}, \ldots\ldots, z_{j,N}$ (IMSL, Inc., 1991):

1. Estimate the mean ($\bar{z}_i$, $\bar{z}_j$) for each time series.

2. Estimate the autocovariance function for each time series.

   The autocovariance function of $\{z_{i,t}\}$, $\sigma_{z_i}(k)$, is estimated by:

$$s_{z_i}(k) = \frac{1}{n} \sum_{t=1}^{n-k} (z_{i,t} - \bar{z}_i)(z_{i,t+k} - \bar{z}_i), \qquad k = 0, 1, \ldots\ldots, K \tag{2.83}$$

   where K = the maximum lag considered

   The autocovariance function of $\{z_{j,t}\}$, $\sigma_{z_j}(k)$, is estimated in similar fashion.

3. Estimate the ACF for each time series.

   The ACF of $\{z_{i,t}\}$, $\rho_{z_i}(k)$, is estimated by:

$$r_{z_i}(k) = \frac{s_{z_i}(k)}{s_{z_i}(0)}, \qquad k = 0, 1, \ldots\ldots, K \tag{2.84}$$

where K = the maximum lag considered

The ACF of $\{z_{j,t}\}$, $\rho_{z_j}(k)$, is estimated in similar fashion.

4. Estimate the cross-covariance function.

The cross-covariance function between $\{z_{i,t}\}$ and $\{z_{j,t}\}$, $\sigma_{z_i z_j}(k)$, is estimated by:

$$s_{z_i z_j}(k) = \begin{cases} \frac{1}{n} \sum_{t=1}^{n-k} (z_{i,t} - \bar{z}_i)(z_{j,t+k} - \bar{z}_j), & k = 0, 1, \ldots, K \\ \frac{1}{n} \sum_{t=1-k}^{n} (z_{i,t} - \bar{z}_i)(z_{j,t+k} - \bar{z}_j), & k = -1, -2, \ldots, -K \end{cases} \quad (2.85)$$

It should be noted that when i = j, the value of $s_{z_i z_j}(k)$ is that of the autocovariance function.

5. Estimate the CCF.

The CCF between $\{z_{i,t}\}$ and $\{z_{j,t}\}$, $\rho_{z_i z_j}(k)$, is estimated by:

$$r_{z_i z_j}(k) = \frac{s_{z_i z_j}(k)}{[s_{z_i}(0)\ s_{z_j}(0)]^{1/2}}, \qquad k = 0, \pm 1, \ldots, \pm K \quad (2.86)$$

It should be noted that when i = j, the value of $r_{z_i z_j}(k)$ is that of the ACF.

For each value of k, the autocovariances and cross-covariances may be formed into a covariance matrix. When considering N' time series, $z_{t,i}$, i = 1, ...,N', Equation (2.85) yields the *ij*th element of the covariance matrix.

In general, the covariance matrix at lag k is not the same as the covariance matrix at lag -k. However, the *ij*th element of the covariance matrix at lag k is equal to the *ji*th element of the covariance matrix at lag -k. In other words, the covariance matrix at lag k is equal to the transpose of the covariance matrix at lag -k (Harvey, 1981).

The correlation matrix is simply a standardised version of the covariance matrix. The *ij*th element of the correlation matrix is given by Equation (2.86).

Tiao and Box (1981) suggest that $\rho_{z_i z_j}(k) = 0$ if the following applies:

$$\left|r_{z_i z_j}\right| \leq 2 \frac{1}{(N-a+1)^{1/2}} \quad \text{for } k > q^c \tag{2.87}$$

2.2.13.4 Stationarity

In order to fit a multivariate time series model of the ARMA class to a number of dependent time series, they need to be jointly stationary (Priestley, 1988). A multivariate time series is jointly stationary if the mean of the series, and all elements of the covariance matrix, are independent of time for all values of k (Brockwell and Davis, 1987). To achieve joint stationarity between component time series, the following steps should be followed.

Firstly, each of the component time series should be checked for stationarity in accordance with the procedure outlined in Section 2.2.6. If the component series are non-stationary, they should be transformed into stationary time series using the differencing procedure described in Section 2.2.7.2. It should be noted that each time series should be differenced the same number of times in order to maintain the phase relationship between them (Harvey, 1981).

However, the conditions for joint stationarity are stronger than just requiring each component series to be stationary. This would simply constrain the diagonal elements of the covariance matrix to be independent of t, without imposing similar constraints on the off-diagonal elements (Harvey, 1981).

As the off-diagonal elements correspond to the cross-covariances between different time series at various lags, the following steps should be followed to ensure that the off-diagonal elements are constrained, and hence the component time series are jointly stationary:

1. The CCF should be calculated between two component series at the lags suggested for ACFs (Section 2.2.6.2).
2. If the CCF cuts off at the lags specified for ACFs in Section 2.2.8.2, the two time series under consideration may be deemed jointly stationary.

If steps 1 and 2 are repeated for all possible combinations of component time series, and each pair of time series is found to be jointly stationary, the multivariate time series may be considered jointly stationary.

2.2.13.5 Model Identification

The model identification process involves choosing the correct model class followed by determining the appropriate order of the AR and MA parameters for the type of model chosen.

Model class

When choosing an appropriate model class, a decision has to be made whether to use a model with constant or periodic parameters (Salas et al., 1985). The dependence relationship between the input and output time series also has to be determined. This can be done in one of the following ways (Hipel, 1985; Salas et al., 1985):

1. Plots of the time series can be inspected in order to get an appreciation of any trends, periodicities, temporal and spatial relationships and the degree of randomness. This information can then be used in conjunction with a sound physical understanding of the data to determine appropriate model classes.

2. Statistical analysis of the data can be carried out. Univariate ARMA models can be fitted to each time series and the cross-correlations between the residuals examined. If the only significant value is at lag zero, the relationship is contemporaneous. If the only significant values occur at lags either greater or smaller than zero, the relationship is unidirectional. If values are significant at positive as well as negative lags, a feedback relationship exists.

Model order

In general, the techniques used for the determination of the order of the AR and MA parameters for multivariate time series models are generalisations of those used in univariate time series analysis (Salas et al., 1985). The model order can be obtained by using an empirical and/or analytical approach (Haltiner and Salas, 1988; Thompstone et al., 1985).

The empirical approach has been successfully used by a number of authors, including Hipel et al. (1977b), Baracos et al. (1981) and McLeod et al. (1983), and involves the utilisation of any knowledge about the physical processes that generated the time series.

The analytical approach is based on the method of Haugh and Box (1977), which makes use of the CCF to determine the appropriate model order. The CCF provides information about the strength of the relationship between time series, as well as the lag at which maximum correspondence occurs (Davis, 1973), and is hence a useful tool for identifying the order of AR parameters to be used in the multivariate model.

The CCF should only be calculated for time series that are jointly stationary (IMSL, Inc., 1991). However, the CCF between two time series not only depends on the relationship between the two time series, but also on the nature of the processes generating the time series (Harvey, 1981). As a result, the two time series under consideration need to be "prewhitened" in some instances. This process involves transforming each component time series into a stationary time series and then fitting a univariate ARIMA model to each time series. The CCF is then calculated for the "whitened" series of residuals (Brockwell and Davis, 1987). This method caters for the case where there is one input series and one output series. In cases where the number of input time series is greater than one, the method can be used to examine the output time series and each of the input time series in turn (Thompstone et al., 1985).

In general, it is difficult to know how many MA parameters are required. Univariate ARIMA models fitted to the component time series may give some guidance, but are by no means conclusive. As a rule, several model configurations have to be tried, and the whiteness of the residuals checked, to see if the proposed model is adequate.

The AIC can also be used as a guide for selecting model orders (Salas et al., 1985; Hipel et al., 1977a).

2.2.13.6 Parameter Estimation

Many authors discuss methods for estimating the unknown model parameters, including Aoki (1987), Schweppe (1973), Harvey (1981, 1989), Lütkepohl (1991) and Christensen (1991). Most of these techniques are based on the principle of maximum likelihood. The advantage of this method is that the parameter estimates obtained are statistically efficient in large samples. However, the disadvantage in using this method is that some kind of iterative procedure is usually required to compute the parameter estimates (Harvey, 1981).

The likelihood function

The joint density function of a set of n observations, $\mathbf{z}_1$, $\mathbf{z}_2$, ...,$\mathbf{z}_n$, is assumed to depend on a set of unknown parameters. Once the observations have been made, the joint density function may be re-interpreted as the likelihood function (Harvey, 1989). In other words, the likelihood function is a measure of the plausibility of a particular set of parameters, given the sample. By maximising the likelihood function, the most plausible set of parameters is obtained.

In the classical sense, the likelihood function applies to independent, identically distributed observations. However, time series analysis deals with observations which are not independent. Despite this, the standard results obtained from the theory of maximum likelihood estimation apply to time series models (Harvey, 1981).

Prediction error decomposition

The prediction error decomposition enables the likelihood function to be broken down into a manageable form (Harvey, 1981). Once the likelihood function has been put into prediction error decomposition form, full maximum likelihood estimation of relatively complex time series models can be carried out. In addition, suitable approximations to full maximum likelihood estimation can be obtained (Harvey, 1981).

The prediction error decomposition form of the log-likelihood function for a multivariate time series model is given below (Harvey, 1981):

$$\log L(\mathbf{z}_1,.....,\mathbf{z}_n) = -\frac{nN'}{2}\log 2\pi - \frac{1}{2}\sum_{t=1}^{n}\log|\Pi_t| - \frac{1}{2}\sum_{t=1}^{n}\mathbf{n}_t'\Pi_t^{-1}\mathbf{n}_t \qquad (2.88)$$

where N' = the number of observations made at each point in time

$\mathbf{n}_t$ = a N'x1 vector of prediction errors at time t with

- mean zero ($E(\mathbf{n}_t) = 0$) and
- covariance matrix Π_t ($Var(\mathbf{n}_t) = \Pi_t$)

A suitable approximation to the log-likelihood function can be obtained by identifying $\mathbf{n}_t$ with the vector of white noise disturbances ($\mathbf{e}_t$) driving the process. If $\mathbf{e}_t$ is normally, identically distributed with mean zero and covariance matrix **H**, then the log-likelihood function can be approximated by:

$$\log L(.) = -\frac{nN'}{2}\log 2\pi - \frac{n}{2}\sum\log|\mathbf{H}| - \frac{1}{2}\sum_{t=1}^{n}\mathbf{e}_t'\mathbf{H}^{-1}\mathbf{e}_t \qquad (2.89)$$

Maximisation of the likelihood function

The best estimates for the unknown model parameters are found by maximising the log-likelihood function. The maximum likelihood estimators are generally obtained by differentiating the log-likelihood function with respect to each of the unknown parameters. By setting these derivatives equal to zero, the likelihood equations are obtained. In most cases, the likelihood equations are non-linear, and the maximum likelihood estimates have to be found using an iterative procedure.

Some commonly used numerical optimisation methods for maximising the log-likelihood function are listed below:

- The EM (expectation step - maximisation step) algorithm proposed by Dempster et al. (1977).
- The scoring algorithm discussed by Lütkepohl (1991).
- The Newton-Raphson algorithm as discussed by Gupta and Mehra (1974), Jones(1980) and Ansley and Kohn (1984).
- The Gauss-Newton method.

2.2.13.7 Diagnostic Checking

The methods for checking the whiteness of the residuals described for univariate time series in Section 2.2.10 can generally be extended to multivariate time series. Plots of the residuals can be examined, and the histograms of their distributions checked for normality and homoscedasticity, as in the univariate case. In addition, the ACFs of the residuals and the CCFs between them should be examined, as they are assumed to be independent of each other (Thompstone et al., 1985).

Harvey (1981) suggests that a Q^{BP} matrix can be obtained as a generalisation of the Box-Pierce chi-square statistic. The diagonal elements of the Q^{BP} matrix consist of the Q^{BP} values calculated for the residuals of the appropriate univariate time series, whereas the off-diagonal elements of the Q^{BP} matrix consist of the Q^{BP} values calculated for the appropriate residual cross-correlations. The Q^{BP} matrix can be used to obtain a rough indication of the goodness of fit of the model by testing each individual matrix component against the chi-square distribution.

2.2.13.8 State Space Models

State space models are a very powerful tool for handling time series data and give a very compact description of any finite parameter linear model (Priestley, 1988). Consequently, state space models provide a general paradigm for modelling multivariate time series (Christensen, 1991).

State space models were first presented by Kalman (1960) and Kalman and Bucy (1960). Originally, the state space representation was introduced to track missiles using satellite observations of their positions, but have since been used extensively in system theory, the physical sciences and engineering (Lütkepohl, 1991).

The general philosophy underlying the state space form is that an observed multivariate time series, $\mathbf{z}_1$, $\mathbf{z}_2$, ...,$\mathbf{z}_N$, depends on a state, φ_t, that may not be observable and is driven by a stochastic process (Lütkepohl, 1991).

Modelling time series in state space form has the following advantages (Harvey, 1989, 1981):

- State space models can cope with missing observations and temporal aggregation.
- The state space representation allows time series models to be formulated in continuous time.
- The state space form can deal with time series that have irregularly spaced observations.
- State space models can model variables that are not directly observable.
- State space models can accommodate variables that are subject to systematic distortion and are contaminated by noise.

Model specification

The general state space model is specified by two equations; the measurement (or observation) equation and the transition (or state) equation.

The measurement equation relates a multivariate time series $\mathbf{z}_t$, containing N' elements, to a m x 1 vector of variables, φ_t, known as the state vector (Harvey, 1989). The variables in the state vector may or may not be observable. The measurement equation is given by (Harvey, 1981, 1984):

$$\mathbf{z}_t = \mathbf{Y}_t \varphi_t + \mathbf{e}_t , \qquad t=1, \ldots\ldots, n \qquad (2.90)$$

where $\mathbf{Y}_t$ = a N' x m matrix

$\mathbf{e}_t$ = a N' x 1 vector of serially uncorrelated disturbances with
- mean zero ($E(\mathbf{e}_t) = 0$) and
- covariance matrix $\mathbf{H}_t$ ($Var(\mathbf{e}_t) = \mathbf{H}_t$)

Despite the fact that the state vector is generally not observable, it is assumed to be driven by a first order Markov process, which is described by the transition equation (Harvey, 1981, 1984):

$$\varphi_t = \mathbf{T}_t \varphi_{t-1} + \mathbf{f}_t, \qquad t=1, \ldots\ldots, n \tag{2.91}$$

where $\mathbf{T}_t$ = a m x m matrix

$\mathbf{f}_t$ = a m x 1 vector of serially uncorrelated disturbances with
- mean zero ($E(\mathbf{f}_t) = 0$) and
- covariance matrix $\mathbf{Q}_t$ ($Var(\mathbf{f}_t) = \mathbf{Q}_t$)

The specification of a linear state space model is completed by the following assumptions (Harvey, 1989, 1981; Lütkepohl, 1991):

1. The initial state vector, φ_0, has a mean of $\mathbf{j}_0$ ($E(\varphi_0) = \mathbf{j}_0$) and a covariance matrix $\mathbf{P}_0$ ($Var(\varphi_0) = \mathbf{P}_0$).
2. The noise terms $\mathbf{e}_t$ and $\mathbf{f}_t$ are uncorrelated with each other at all times and are also uncorrelated with the initial state, φ_0.

When specifying a model in state space form, the following objectives should be met (Harvey, 1989; Priestley, 1988):

1. The state vector, φ_t, should contain all the relevant information about the system at time t.
2. The state vector should not contain any redundant variables. If this is the case, the equations describing the state space model are said to provide minimal realisation, and the dimension of the state vector may be called the dimension of the model.

System matrices (Harvey, 1989; Lütkepohl, 1991)

The system matrices of a state space model are defined as the unknown matrices in the measurement equation ($\mathbf{Y}_t$ and $\mathbf{H}_t$) and the unknown matrices in the transition equation ($\mathbf{T}_t$ and $\mathbf{Q}_t$). The system matrices are assumed to be non-stochastic. Consequently, if they change with time, they do so in a manner which is predetermined (Harvey, 1989), resulting in a linear system in which $\mathbf{z}_t$ can be expressed as a linear combination of past and present disturbances ($\mathbf{e}_t$ and $\mathbf{f}_t$) and the initial state vector (φ_0).

Frequently, the system matrices do not change with time, in which case the model is referred to as time-invariant or time homogeneous. For this class of models, the state space equations can be written as:

$$\mathbf{z}_t = \mathbf{Y}\boldsymbol{\varphi}_t + \mathbf{e}_t, \qquad \mathrm{Var}(\mathbf{e}_t) = \mathbf{H} \tag{2.92}$$

$$\boldsymbol{\varphi}_t = \mathbf{T}\boldsymbol{\varphi}_{t-1} + \mathbf{f}_t, \qquad \mathrm{Var}(\mathbf{f}_t) = \mathbf{Q} \tag{2.93}$$

When specifying a state space model, the system matrices are assumed to be known. However, this is usually not the case, and in many instances the main statistical task is to estimate these unknown matrices. The unknown model parameters are called hyperparameters and determine the stochastic properties of the model. They are also the parameters on which the likelihood function depends (Lütkepohl, 1991).

VARMA models in state space form

The fact that the state equation is a first order AR process might appear to be rather restrictive. However, since it is a vector process, there is nothing fundamentally first order about it (Christensen, 1991). In fact, any finite-order linear differential or difference equation can be expressed as a vector first order equation (Priestley, 1988). Consequently, any VARMA model can be represented in state space form.

A. The VAR (q) process (Lütkepohl, 1991)

All finite order VAR models can be cast into state space form. The general VAR (p) process is given by:

$$\mathbf{z}_t = \Phi_1\mathbf{z}_{t-1} + \ldots\ldots + \Phi_p\mathbf{z}_{t-p} + \mathbf{e}_t \tag{2.94}$$

where
- $\mathbf{z}_t$ = a N' x 1 vector of observations
- Φ_i = a N' x N' matrix of AR parameters
- $\mathbf{e}_t$ = a N' x 1 vector of serially uncorrelated disturbances

One possible state space representation for the VAR (p) process is given below. The following notation applies:

$$\mathbf{Z}_t = \begin{bmatrix} \mathbf{z}_t \\ . \\ . \\ . \\ \mathbf{z}_{t-p+1} \end{bmatrix}, \quad \Phi = \begin{bmatrix} \Phi_1 & .... & \Phi_{p-1} & \Phi_p \\ \mathbf{I}_k & & 0 & 0 \\ & & . & . \\ & & . & . \\ & & . & . \\ 0 & .... & \mathbf{I}_k & 0 \end{bmatrix} \text{ and } \mathbf{E}_t = \begin{bmatrix} \mathbf{e}_t \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}$$

$\mathbf{I}_k$ = the identity matrix

The transition equation is given by:

$$\mathbf{Z}_t = \Phi \mathbf{Z}_{t-1} + \mathbf{E}_t \tag{2.95}$$

The measurement equation is given by:

$$\mathbf{z}_t = [\mathbf{I}_k \ 0 \ ... \ 0] \mathbf{Z}_t \tag{2.96}$$

When comparing Equations (2.96) and (2.95) with Equations (2.90) and (2.91) it can be seen that in this case:

$\mathbf{z}_t = \mathbf{z}_t$
$\varphi_t = \mathbf{Z}_t$
$\mathbf{Y}_t = [\mathbf{I}_k \ 0 \ ... \ 0]$
$\mathbf{e}_t = 0$
$\mathbf{T}_t = \Phi$
$\mathbf{f}_t = \mathbf{E}_t$

B. The VARMA (p,q) process (Lütkepohl, 1991)

The general VARMA (p,q) process is given by Equation (2.81). One possible state space representation of the general VARMA process is obtained by choosing the following notation:

$$\mathbf{Z}_t = \begin{bmatrix} z_t \\ . \\ . \\ . \\ z_{t-p+1} \\ e_t \\ . \\ . \\ . \\ e_{t-q+1} \end{bmatrix} \qquad \mathbf{E}_t = \begin{bmatrix} e_{t+1} \\ . \\ . \\ . \\ 0 \\ e_{t+1} \\ 0 \\ . \\ . \\ . \\ 0 \end{bmatrix}$$

$$\Phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}$$

where

$$\Phi_{11} = \begin{bmatrix} \Phi_1 & .... & \Phi_{p-1} & \Phi_p \\ I_k & & 0 & 0 \\ & & . & . \\ & & . & . \\ & & . & . \\ 0 & .... & I_k & 0 \end{bmatrix}, \quad \Phi_{12} = \begin{bmatrix} \Theta_1 & .... & \Theta_{q-1} & \Theta_q \\ 0 & & 0 & 0 \\ & & . & . \\ & & . & . \\ & & . & . \\ 0 & .... & 0 & 0 \end{bmatrix}, \quad \Phi_{21} = 0$$

and

$$\Phi_{22} = \begin{bmatrix} 0 & .... & 0 & 0 \\ I_k & & 0 & 0 \\ & & . & . \\ & & . & . \\ & & . & . \\ 0 & .... & I_k & 0 \end{bmatrix}$$

The transition and measurement equations are given by Equations (2.95) and (2.96) respectively.

In general, there is no unique state space representation of a linear model. The state vector and the system matrices can usually take on a variety of forms. Other state space

representations of VARMA models are given by Aoki (1987), Hannan and Deistler (1987) and Wei (1990).

When multivariate VAR or VARMA models are cast into state space form, the system matrices **Y** and **T** are time invariant. In addition, the state vector contains at least some observed or observable variables (Lütkepohl, 1991). This is contrary to the traditional use of state space models, where the main objective is the prediction of the unobservable state vector.

2.2.13.9 The Kalman Filter

The Kalman filter is a statistical algorithm for dealing with state space models and was introduced by Kalman (1960) and Kalman and Bucy (1961). The Kalman filter can be used to recursively estimate the state vector, φ_t, as new observations become available. Recursive techniques have the advantage that they:

- Are not as computationally intensive as other techniques.
- Do not require as much memory as other methods.

This is achieved by utilising previous calculations and by retaining in memory only those items necessary for the processing of future observations (Sallas and Harville, 1981).

The Kalman filter can also be used to estimate the hyperparameters with the aid of maximum likelihood estimation. This is because the prediction error decomposition form of the likelihood function is expressed in terms of one-step-ahead prediction errors, which are a by-product of the Kalman filter recursions (Harvey, 1989). By using the prediction error decomposition form of the likelihood function, any unknown model parameters can be estimated. This technique also provides the basis for statistical testing and model specification (Harvey, 1989).

The general form of the Kalman filter

Once a model has been put into state space form, the Kalman filter provides a recursive algorithm for calculating the optimal estimator of the state vector at time t, based on the observations up to and including $\mathbf{z}_t$. This estimate can be continually updated as new observations become available. In order to calculate the estimates of the state vector, the system matrices, as well as the initial conditions, are assumed to be known at each time period.

The general form of the Kalman filter consists of two steps; a prediction step and an updating step. In the prediction step, the optimal estimator of the state vector ($\mathbf{j}_t$) is obtained, based on the information currently available. Once the new observation, $\mathbf{z}_t$, becomes available, the estimator of φ_t obtained in the prediction step is updated.

In general, the recursions are carried out by the procedure given below:

1. Initial values of the state vector and the covariance matrix are specified.
2. The prediction step is carried out at t = 1.
3. The updating step is carried out at t = 1.
4. The prediction and updating steps are then repeated for t = 2, t = 3, ..., t = n.

When all n observations have been processed by the Kalman filter, the optimal estimator of the state vector, based on all observations, is obtained. This opens the way for obtaining optimal predictions of the state vector and of the observations at time t > n (forecasting), and extracting optimal estimators of the state vectors at time t < n based on the entire information set (smoothing).

Using state space models in conjunction with the Kalman filter recursions has the following advantages:

- On-line estimation of the model parameters is possible. As new data become available, the model parameters can be updated using only the most recent information. In other words, there is no need to process the entire data set, which is computationally efficient (Bergman and Delleur, 1985a).

- System noise can be separated from observation noise. The measurement equation caters for observation noise while the transition equation allows system noise to be modelled (Bergman and Delleur, 1985a).

- Data irregularities, such as missing observations, temporal aggregations, irregularly spaced observations and data revisions can be taken into account (Harvey, 1984). In order to deal with missing observations, the appropriate updating equations have to be omitted as part of the Kalman filter recursions. A maximum mean squared estimator (MMSE) of the missing observation can then be obtained with the aid of a smoothing filter once all the observations have been processed (Harvey, 1984). It should be noted that the prediction error decomposition form of the ML function can still be obtained when some observations are missing. The above procedure also caters for outliers, which can be omitted from the data set and processed as missing observations.

2.2.13.10 Time Variable Parameter (TVP) Models

Recently, more advanced time series analysis procedures have been developed, which are able to fit models to non-stationary data. In these models, the model parameters vary with time, resulting in Time Variable Parameter (TVP) models (Young et al., 1991). TVP models are capable of modelling gradual as well as sudden fluctuations in the data, including discontinuities (Ng and Young, 1990).

Generally, TVP models are written in state space form and make use of optimal state estimation algorithms such as the Kalman filter (Young, 1994). In TVP modelling, the Kalman filter is not used to represent the stochastic model itself, but to model the variations of the model parameters (Young, 1994).

The state space representation and the Kalman filter can be most effectively utilised when TVP models are used in structural time series form (Harvey, 1984; Harvey and Peters, 1990). Structural time series models provide a general framework in which different types of time series models can be modelled (Harvey, 1984). They consist of trend, seasonal and random noise components (Equation 2.45), which are modelled directly. The model is therefore structural, as each of the individual components can be given direct interpretation (Harvey, 1984).

Generally, a state space approach is used to model the seasonal and trend components. The majority of the component models are based on a simple Generalised Random Walk (GRW) approach, which caters for non-stationarities in the states as well as the parameters (Ng and Young, 1990). The individual component models are then combined, resulting in a discrete-time state space representation of the observed time series (Ng and Young, 1990). The Kalman filter can then be used for recursive state estimation, forecasting and smoothing (Young et al., 1991).

The model identification and parameter estimation stages in the development of structural models are by no means trivial. The first attempt to address this problem was made by Harrison and Stevens (1971, 1976), who used a Bayesian approach. This method involves fixing the variance parameters (e.g. the variance of the white noise terms in the state space equations), on which the forecasts obtained using structural models depend, using *a priori* information, rather than estimating them (Harvey and Peters, 1990). When using the Bayesian model, the role of the Kalman filter is to obtain an estimate of a posterior distribution for the state vector by updating a prior distribution (Harvey, 1984). Consequently, $\mathbf{j}_t$ is not the MMSE of φ_t, but the mean of

the posterior distribution of φ_t (Harvey, 1984). One of the advantages of the Bayesian approach is that any *a priori* knowledge can be built into the model. However, the fact that the variance parameters have to be chosen *a priori* can also be restrictive, as sufficient knowledge to do this successfully may not always be available in practice.

The need for any *a priori* knowledge can be minimised by formulating structural models in ML terms (Harvey, 1984; Harvey and Peters, 1990). In this approach, the likelihood function is obtained in prediction error decomposition form with the aid of the Kalman filter. The model parameters are then estimated by maximising the likelihood function using an appropriate numerical optimisation method. However, this procedure is quite complex, even for relatively simple models (Ng and Young, 1990). In addition, the above method makes it "difficult to formulate all aspects of parameter and state estimation in completely recursive terms, so restricting adaptive implementation of the resulting algorithms" (Ng and Young, 1990).

To simplify the model identification and parameter estimation phases of model development, a Sequential Spectral Decomposition (SSD) procedure has been proposed (Young, 1988; Young et al., 1989; Ng and Young, 1990; Young et al., 1991; Young, 1994). This procedure also fully utilises recursive solutions at all stages of modelling, thus catering for all types of non-stationarity (Ng and Young, 1990). The SSD approach is used to "sequentially decompose the time series into quasi-orthogonal estimates of the trend and periodic components" (Young et al., 1991). The first step in this procedure is to estimate the trend component using a stochastic, second-order, Integrated Random Walk (IRW) model. The trend component is then subtracted, and the remaining time series modelled using a General Transfer Function (GTF) or Dynamic Harmonic Regression (DHR) model. As a result, the non-linear estimation procedure is broken down into a number of linear parts, each of which is solved in fully recursive terms (Young et al., 1991).

2.2.13.11 Forecasting

The forecasting procedure for multivariate models is very similar to that for univariate time series. Forecasting involves the extrapolation of the components estimated at the end of the sample (Harvey, 1989). The larger the forecast time span, the greater the uncertainty of the forecast. Forecasts will only lie between the specified confidence limits if the fitted model continues to be an adequate representation of the underlying data generating process (Harvey, 1989).

2.2.13.12 Applications

Time series analysis has been widely used in the field of economics (Nelson, 1973; Granger and Newbold, 1977; Granger, 1980; Abraham and Ledolter, 1983; Pankratz, 1983; Abraham, 1993).

Time series analysis methods have also been extensively used to model hydrological time series, water quality time series, water demand, water pricing and meteorological time series, and are a vital tool used in water resources planning and management (Hipel, 1985).

In the field of water resources engineering, time series analysis is used for forecasting, generating synthetic data sequences and investigating and modelling the underlying processes of the system of interest (Hipel, 1985). Examples of specific applications of time series models of the ARMA type for forecasting water resources time series are given below.

Modelling River Flows

Salas et al. (1985) developed CARMA and VARMA models for modelling the annual river flows of the Green River near Palmer and the South Fork Skykomish River near Index, both located in the State of Washington, U.S.A. Both models performed well, but the CARMA model was preferred from a practical viewpoint, as it was more parsimonious and its model parameters were easier to estimate.

Camacho et al. (1985) developed a CARMA model for modelling annual riverflows for the Wolf River near London, Wisconsin, and the Fox River near Berlin, Wisconsin.

Vecchia (1985) developed a PARMA model to model monthly streamflows for the Rio Caroni River in Venezuela.

Haltiner and Salas (1988) developed an ARMAX model to forecast daily streamflows in the headwaters of the Rio Grande River in southern Colorado, U.S.A. The forecasting periods considered include one and three days. Daily precipitation, streamflow, average daily temperature and snow-covered-area data were used as model inputs.

Thompstone et al. (1985) developed a number of time series models to obtain one-step forecasts of quarter-monthly (i.e. near weekly) natural inflows to the Lac St. Jean

Reservoir, Quebec, Canada. ARMA and PARMA models were fitted to the deseasonalised, log-transformed inflow time series. In addition, transfer function models were developed, using the inflow as well as rainfall and snowmelt time series as inputs. The results obtained were compared with those produced by a physically based model simulating the relationship between daily meteorological conditions and natural inflows to various reservoirs. The best performance was obtained using the transfer function model, followed by the ARMA model, the PAR model and the physically based model.

Bergman and Delleur (1985b) developed an AR model in state space form to obtain forecasts of daily river flows for the Potomac River near Pointy Rocks, Maryland, U.S.A. The Kalman filter was used to update the model parameters in real time. The maximum forecasting period used was four days.

Modelling Water Levels

Irvine and Eberhardt (1992) developed a univariate multiplicative seasonal ARIMA model for forecasting mean monthly water levels for Lakes Erie and Ontario, Canada, up to six months in advance. Although the model was able to forecast the general trends in the data for all forecasting periods, a maximum forecasting length of three months was found to be desirable for greatest accuracy.

Modelling Water Quality Parameters

Camacho et al. (1985) developed a CARMA model to model monthly concentrations of total and Kjeldahl nitrogen in the Middle Fork Creek near Seebe, Alberta, Canada.

2.3 Artificial Neural Networks for Modelling Time Series

Back-propagation networks are the neural network type most suited to forecasting (NeuralWare, Inc., 1991). Examples of the successful application of backpropagation networks for forecasting have been discussed in Section 2.1.9.5. An advantage of back-propagation networks, when compared with more complex network types, is that they are not as computationally intensive and hence take less time to train (Chakraborty, 1992).

In this section, the use of back-propagation networks is compared with traditional methods of time series analysis. The class of ARMA models has been chosen as the

basis of comparison as they are "... accepted, standard representations of stochastic time series" (Young et al., 1991) and are among those forecasting models most successfully applied in practice (Tang et al., 1991).

When using ANNs for time series analysis, the modelling philosophy employed is similar to that used in traditional statistical approaches. In both cases, the unknown model parameters (i.e. the connection weights in the case of the ANN) are adjusted in order to obtain the best match between a historical set of model inputs and the corresponding outputs. The steps involved in developing an ANN model are outlined in Figure 2.18.

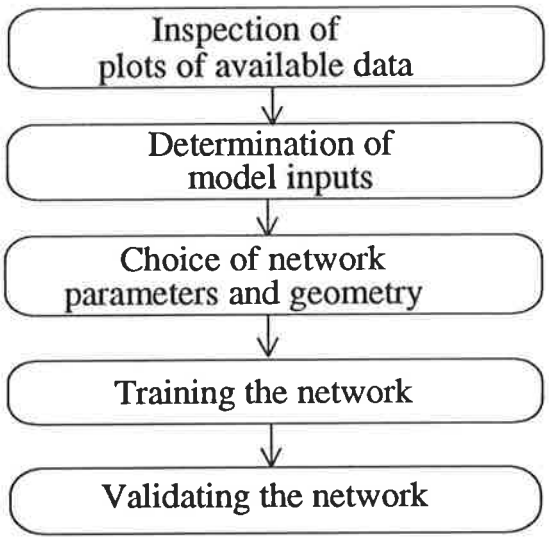


Figure 2.18: Steps in the Development of an ANN model

2.3.1 Inspection of Plots of the Time Series

As discussed in Section 2.2.6.1, data familiarisation should be the first step in the development of any time series model. Inspecting the data plots gives an indication of any trends or seasonal variation in the data and enables the detection of outliers, discontinuities and turning points.

2.3.2 Determination of the Model Inputs

One of the advantages of using a neural network approach for modelling time series is that the amount of data pre-processing required is minimal.

In order to fit an ARMA type model to a time series, or a set of time series, the data are assumed to follow a normal distribution (Irvine and Eberhardt, 1992). If this is not the

case, the data need to be transformed using the Box-Cox transformation (Box and Cox, 1964). In contrast, there is no need to transform the data when developing an ANN model, regardless of the distribution of the data (NeuralWare, Inc., 1991; Burke and Ignizio, 1992).

More recent time series approaches enable models to be fitted to time series that follow the Exponential, Gamma, Weibull, Laplace, Beta or Mixed Exponential distributions (Lewis, 1985). However, in order to use these models, the exact distribution of the data still has to be known.

When developing ANN models, the raw data are usually used without transforming them into stationary data first, as ANNs have the ability to determine underlying relationships in the data, such as trends and seasonal variations, with the aid of their hidden layer nodes. However, to the author's knowledge, the effect of using stationary data, as opposed to the raw data, has not yet been investigated. Such a comparison is carried out in Section 3.4.

When developing ARMA type models, on the other hand, the data used have to be stationary. If this is not the case, the data have to be transformed using techniques such as differencing (Box and Jenkins, 1976). Recently the use of TVP models has been suggested as a means of dealing with non-stationary data directly (Young, 1990; Ng and Young, 1990; Young, 1994).

ANNs belong to the class of data driven approaches (Chakraborty et al., 1992), which, when presented with a set of input and corresponding output data, have the ability to determine which of the model inputs are critical (Daniell, 1991; NeuralWare, Inc., 1991). The forecasting ability of ANNs is not compromised by the presence of non-critical inputs, suggesting that ANNs are not significantly affected by model noise. In fact, Tang et al. (1991) found that increasing the number of model inputs marginally improved the forecasts obtained, as the ANN was able to make use of any additional information presented to it. The above properties make ANN models well suited for complex problems, "which may be poorly understood and which cannot be adequately described by a set of rules or equations" (Maren et al., 1990).

The class of ARMA models, on the other hand, belong to the class of model driven approaches (Chakraborty, 1992). In this type of model, the structure of the model is determined first, before the unknown model parameters are estimated.

Despite the fact that ANNs are data driven, it is desirable to have some indication as to which inputs should be included in the model, especially when dealing with complex, multivariate time series where the number of potential inputs could be in the order of a few hundred. Having a large number of inputs generally increases the size of the network, resulting in increased training times, possible over-parameterisation and poor local minima.

To the author's knowledge, no analytical approaches for determining the inputs for multivariate ANN models have been developed to date. Lachtermacher (1993) and Lachtermacher and Fuller (1994) have developed a hybrid methodology for determining the inputs for univariate ANN models. This method involves fitting a univariate Box-Jenkins type model to the time series, which is then used as a guide for determining the inputs to the ANN model. However, this method is only applicable to univariate models and has not yet been extended to cater for multivariate cases. In this research, the problem of determining inputs for multivariate ANN models is addressed in Section 3.10.

2.3.3 Choice of the Network Parameters and Geometry

One of the disadvantages of back-propagation models is that the internal model parameters (e.g. learning rate, momentum and epoch size) and the network geometry (i.e. the number of hidden layers and the number of nodes per hidden layer) have to be determined by trial and error, although some guidance is available, as discussed in Section 2.1.8.3.

Users of backpropagation networks are often unsure of the effect different internal parameters and network geometries have on model performance. A detailed sensitivity analysis was carried out in this research (Section 3.5.3) to demonstrate the effect of various internal parameters and network geometries on training speed and generalisation ability, and to devise some general guidelines for optimising network performance.

2.3.4 Training of the Network

The training data are usually presented to the network repeatedly until the stopping criteria chosen (Section 2.1.8.1) are satisfied. However, it should be noted that the number of training samples that need to be presented to the network to achieve a certain performance level can vary greatly, depending on the internal network parameters used.

ANNs generally learn an approximation to the underlying rules governing a relationship (NeuralWare, Inc., 1991). Consequently, they are not well suited to learning deterministic input - output relationships, as they are usually not able to perform exact mappings. However, because ANNs have the ability to learn the underlying relationships between inputs and outputs (as opposed to the exact relationship), they are relatively insensitive to noise and have good generalisation ability (NeuralWare, Inc., 1991; Hubick, 1992; Maren et al., 1990; Tang et al., 1991; Burke and Ignizio, 1992). However, care must be taken not to include an excessive number of connection weights, as this might result in overfitting of the data (Burke and Ignizio, 1992), which is undesirable. In contrast, ARMA type models are sensitive to noise (Tang et al., 1991). However, data noise can be modelled explicitly by transforming ARMA models into state space form and using the Kalman filter.

ANN models also perform better than ARMA type models when a limited data record is available (Tang et al., 1991; Schizas et al., 1994). This is of great importance when dealing with real life problems, as the amount of data available is often constrained by financial considerations.

2.3.5 Validation of the Network

When validating an ANN model, the traditional diagnostic checking step of examining the residuals is usually not carried out. The adequacy of an ANN model is generally determined by assessing the forecasting capability of the model when presented with a data set not used in the training phase.

2.3.6 Forecasting

The ARMA class of models is suited to shorter term forecasting (Tang et al., 1991). However, ARMA type models are not suited to longer term forecasting, as multi-step forecasts are obtained in a recursive fashion (Lachtermacher, 1993; Tang et al., 1991). ANN models, on the other hand, are well suited to longer term forecasting, as they base their forecasts on the approximate underlying relationships in the data (Tang et al., 1991; NeuralWare, Inc., 1991).

However, when forecasting for more than one time step, it is unclear whether it is better to train ANNs for a single time step forecast and then apply it recursively, or to train them for a multi-step forecast directly. Lapedes and Farber (1988) found the former method to be superior in obtaining multi-step forecasts of a noise-free, chaotic

time series. On the other hand, Tang et al. (1991) had great success in obtaining longer-term forecasts of domestic car sales in the U.S.A. using the direct forecasting method. The question of whether it is better to use the direct or recursive forecasting method is addressed in Section 3.4.

Several researchers have compared the performance of ANN models with ARMA type models (e.g. Chakraborty, 1992; Tang et al., 1991; Lachtermacher and Fuller, 1994). However, to date, the majority of these comparisons have been restricted to simple univariate models (e.g. Tang et al., 1991; Lachtermacher and Fuller, 1994). Tang et al. (1991) suggest that the comparison of ANN models with more conventional models should be extended to the multivariate case. Chakraborty et al. (1992) compared the performance trivariate ANN and ARMA type models relating flower prices for the cities of Buffalo, Minneapolis and Kansas City. However, to the author's knowledge, no comparison has been carried out for multivariate models using causal variables, especially in the field of water resources engineering. Such a comparison is carried out in Section 3.8.

2.3.7 Other Differences Between ANN and ARMA Type Models

Although some time series models cater for non-linearities (Tong et al., 1985; Ozaki, 1985; Brillinger, 1985), models of the ARMA type are usually linear. This makes the modelling of natural systems, which are generally highly non-linear, very difficult. A detailed discussion of the disadvantages of non-linear models is given by Tong (1983). ANNs overcome this problem with the aid of their non-linear transfer functions (Burke and Ignizio, 1992).

One of the disadvantages of ANN models is that the relationship between variables cannot be obtained directly. However, information about the relative significance of the various inputs can still be obtained by carrying out a sensitivity analysis, as demonstrated in Chapters 3 and 4.

An advantage of ANN methods is that they require less storage and are much faster than conventional statistical approaches (Burke and Ignizio, 1992). At present, the processing speed of ANNs is limited by the lack of readily available parallel computers. However, with advances in technology, this situation is likely to be rectified in the future, allowing the full capabilities of ANNs to be exploited.

Chapter 3

Salinity Forecasting in the River Murray

3.1 Introduction

Human induced salinisation is an increasing cause of environmental problems, especially in the arid and semi-arid regions of the world (Ghassemi et al., 1991). The South Australian portion of the Murray-Darling Basin in the southeast of Australia is one such region. Increasing levels of salinity in the River Murray as a result of land clearance, inefficient irrigation practices and flow regulation have had a number of adverse effects on water users. The River Murray supplies water for irrigation and for cities in South Australia which have access to the river via pipelines, including Adelaide (population 1 million). The high salinity of the water results in corrosion of pipes and fittings, increased usage of soap and detergents, as well as soil salinisation and a reduction in crop yield. It is estimated that high levels of salinity cause $30 million damage per year to domestic and industrial users in Adelaide (Dwyer Leslie Pty. Ltd., 1984).

In this chapter, univariate and multivariate artificial neural network (ANN) and time series models of the ARIMA type are developed for forecasting salinity in the River Murray at Murray Bridge. A significant part of Adelaide's water supply is abstracted at this point of the river. Forecasts of salinity can be used to modify pumping schedules so that less water is pumped at times of high salinity and more water is pumped at times of

low salinity. Dandy and Crawley (1992) have developed a model for optimising pumping policies taking salinity into account. Results from the model indicate that the average salinity of the water supplied to Adelaide could be reduced by about 10% if salinity was forecast several weeks in advance and pumping policies were modified accordingly.

The objectives of this chapter are to:

1. Develop models aimed at obtaining the best possible forecasts of salinity in the River Murray at Murray Bridge.

2. Use the case study of salinity forecasting to investigate the following issues in relation to the modelling of time series using artificial neural networks (ANNs):
   - What is the effect of adding less significant parameters on network performance?
   - When forecasting for more than one time step, is it better to train the network for a single time step forecast and then apply it recursively or train it for a multistep forecast directly?
   - Is it better to train networks using stationary data or raw data?
   - What is the effect of including hidden layers?
   - Are there any methods for determining inputs for multivariate ANN models?
   - What is the effect of internal parameters and geometry on network performance?
   - How does the training data used affect network performance?
   - What is the effect of having multiple network outputs?
   - How does a network behave when it approaches a local minimum in the error surface?
   - Can ANNs be used to extract information about the relationship between the input and output variables?
   - How does the performance of univariate and multivariate ANN models compare with that of the more traditional and widely used time series models of the ARMA type for various forecasting periods?

3.2 Background

The River Murray starts as a small stream near Mt. Kosciusko, New South Wales. For the first 400 km, it flows rapidly and is fed by high rainfalls and melting snow. Subsequently, the flow slows considerably as the river broadens and starts to meander. The water has travelled about 2500 km by the time the river discharges into the sea near Goolwa, South Australia. Along the way, the Murray is joined by the Darling and Murrumbidgee Rivers as well as many small tributaries, and traverses five distinct

regions including the Headwaters, the Riverine Plains, the Mallee Trench, the Mallee Gorge and the Lakes and Coorong. The location of the River Murray is shown in Figure 3.1. The Murray-Darling system is the fourth longest river system in the world, but the volume of water that travels down the river system is low by world standards. Its average annual flow is approximately 16% of that of the Nile, 3.5% of the Mississippi and 0.25% of the Amazon (Mackay and Eastburn, 1990).

The flow in the River Murray is highly regulated by four major storages, a series of 14 locks and weirs and 5 barrages (Figure 3.2). The river is regulated in order to stabilise water levels for irrigation, water supply and navigation purposes. The barrages stop seawater from travelling upstream during times of low flow (Figure 3.3). There are three major storages on the River Murray, namely: Lake Victoria (680 GL), Hume Reservoir (3040 GL) and Dartmouth Reservoir (4000 GL). The fourth major storage is Menindee Lakes (1680 GL), which is located on the Darling River. The combined capacity of the four major storages is 9400 GL.

In addition to these four major storages, there are a number of smaller storages on the Murray's tributaries (Mackay and Eastburn, 1990), including:

- Glenlyon and Beardmore Dams in Queensland.
- Burredong, Wyangala, Carcoar, Burrinjuck, Blowering, Windamere, Keepit, Chaffey, Pindari and Copeton Dams in New South Wales.
- Eildon and Eppalock Dams in Victoria.

The high degree of regulation of the Murray River system is highlighted by the fact that only two of the Murray's tributaries, namely the Kiewa and Ovens Rivers in north eastern Victoria, have not been significantly regulated (Mackay and Eastburn, 1990).

A water sharing agreement between the states determines how River Murray water is divided between New South Wales, Victoria and South Australia, and features the following main points (Mackay and Eastburn, 1990):

1. The flow at Albury (which is located just downstream of the Hume Reservoir, Figure 3.2) is shared equally between New South Wales and Victoria.
2. Victoria and New South Wales retain control of their tributaries below Albury.
3. Victoria and New South Wales supply South Australia with a guaranteed minimum quantity of water, or "entitlement" (Table 3.1). These entitlement flows provide Adelaide with a guaranteed water supply, even in the driest years.

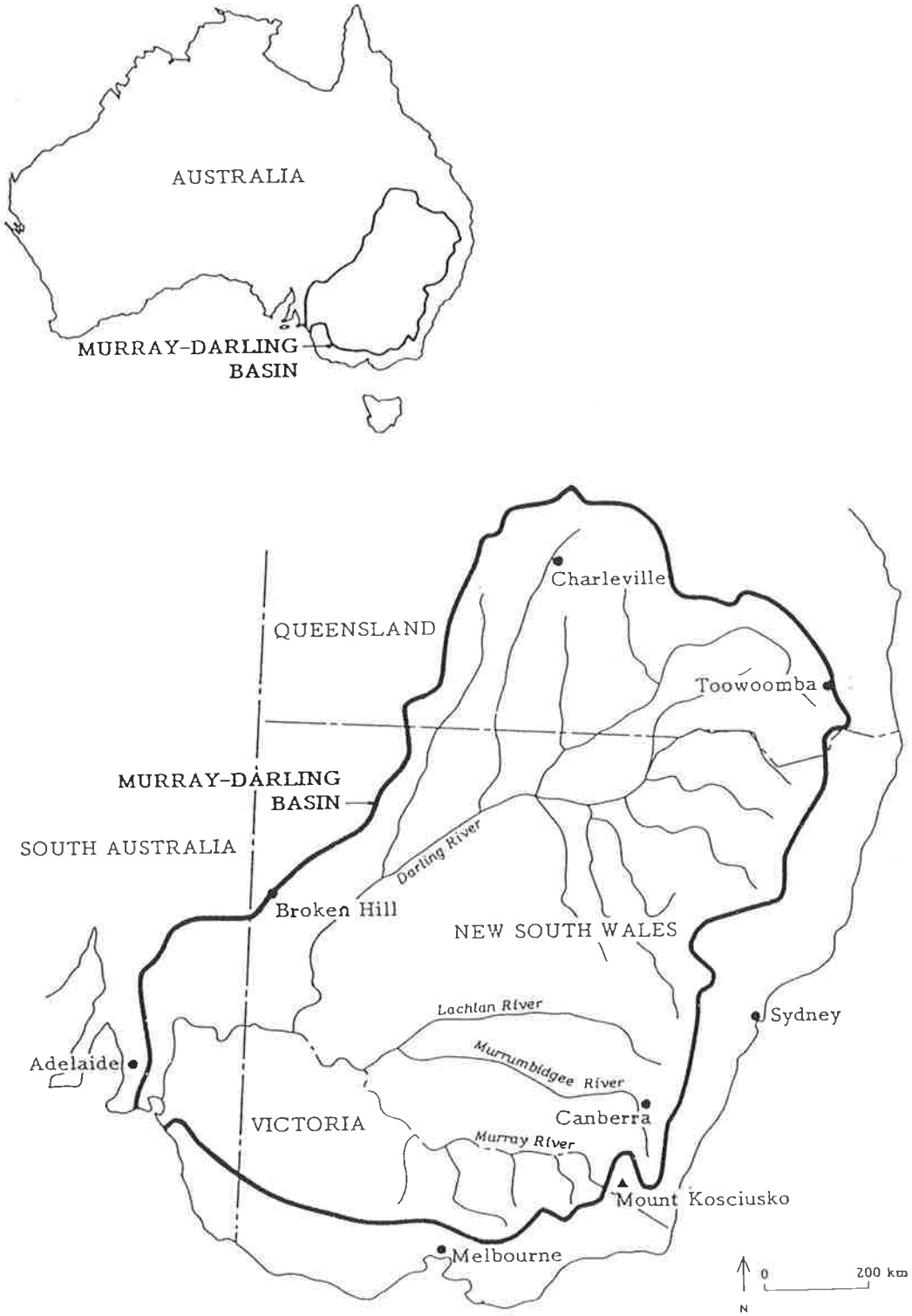


Figure 3.1: Extent of the Murray Darling Basin (Source: Engineering and Water Supply Department (EWS), 1987)

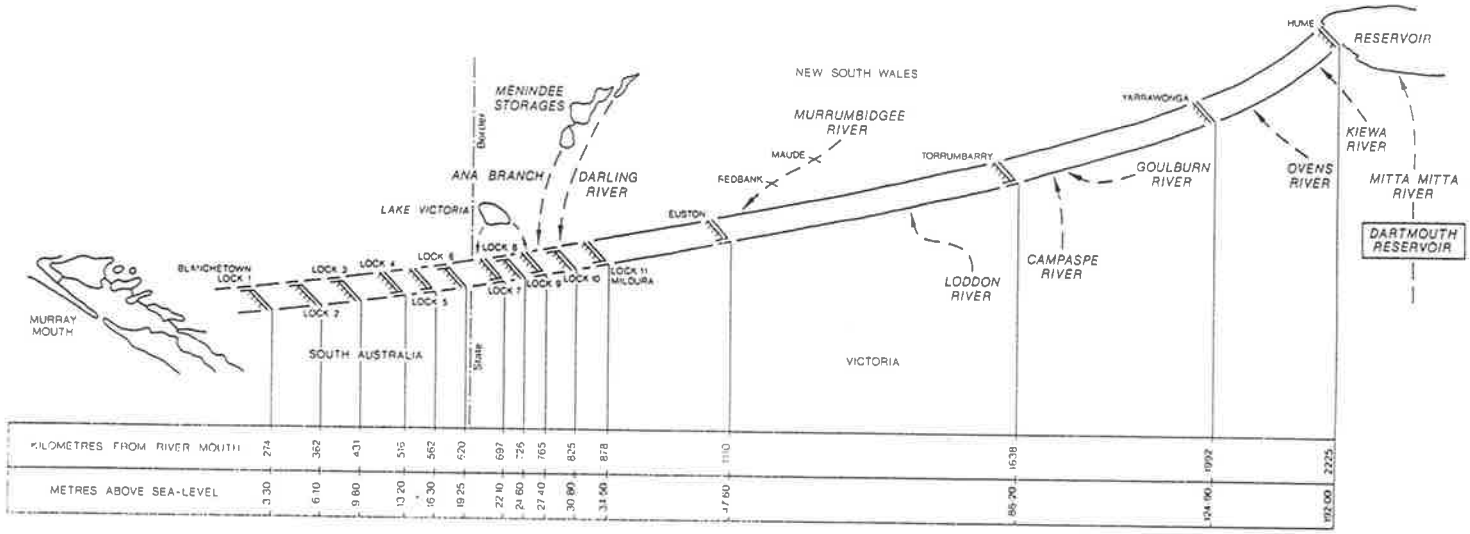


Figure 3.2: Profile of the River Murray (Source: EWS, 1989)

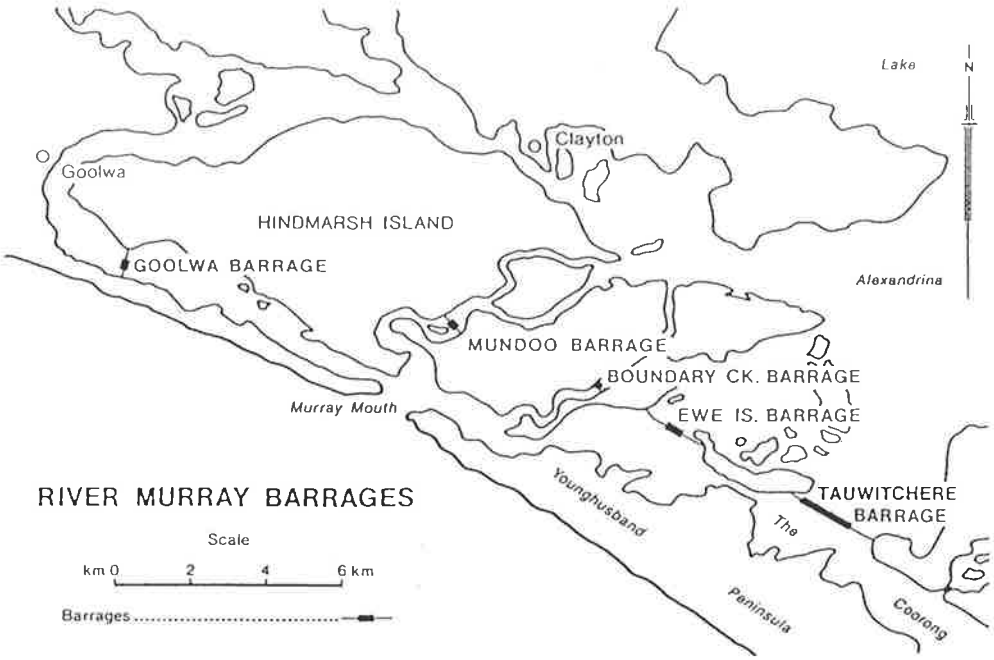


Figure 3.3: The River Murray Barrages (Source: EWS, 1989)

Table 3.1: Entitlement Flows to South Australia

Month	Entitlement Flow (ML/day)
January	7,000
February	6,900
March	6,000
April	4,500
May	3,000
June	3,000
July	3,500
August	4,000
September	4,500
October	5,500
November	6,000
December	7,000

The Murray Darling basin covers 1.06 M km^2, which is one seventh of Australia's surface area (Figure 3.1). There are four major climatic zones throughout the basin. The climate is subtropical in the north, cool and humid in the eastern highlands,

temperate in the south and hot and dry in the west. The mean average rainfall in the catchment is 430 mm and varies from 250 mm in South Australia to 1500 mm in the upper catchment (Ghassemi et al., 1991).

The River Murray is Australia's major surface water resource. It supplies a significant portion of the domestic water for towns along the river and those places in South Australia which have access to River Murray water via pipelines, including Adelaide, Port Pirie, Whyalla, Port Augusta and smaller country towns on the Yorke Peninsula, as well as in the centre and in the south-east of South Australia (Mackay and Eastburn, 1990). The various River Murray pipelines are shown in Figure 3.4.

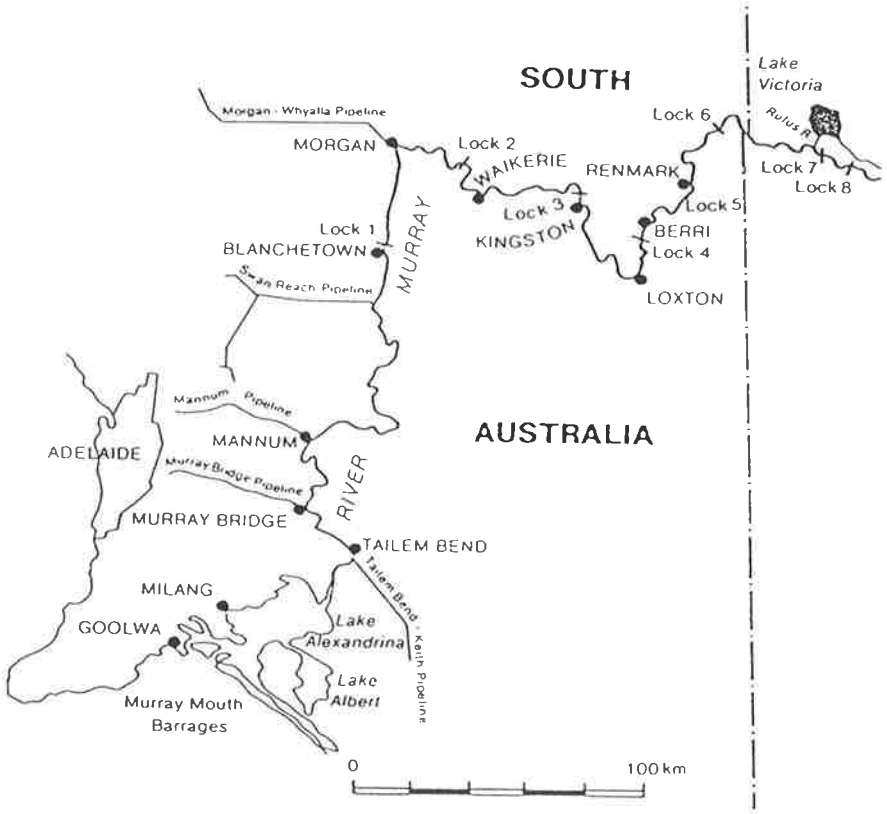


Figure 3.4: The River Murray Pipelines (Source: Mackay and Eastburn, 1990)

The water pumped from the River Murray via the Mannum to Adelaide and Murray Bridge to Onkaparinga pipelines plays a vital role in meeting Adelaide's water demand. Adelaide has a population of approximately 1 million and is the capital of South Australia, which is frequently referred to as the driest state in the driest inhabited continent on earth (Figure 3.1). Although the majority of Adelaide's water supply comes from catchments in the Mt. Lofty Ranges, the amount of water supplied by the River Murray is significant. The fraction of water supplied by the Murray varies

between 10% and 80% in any particular year, with an average of 35% (Dandy and Crawley, 1992). A detailed description of the Adelaide water supply system is given by Dandy and Crawley (1992).

The water obtained from the Mt. Lofty Ranges and the River Murray is of poor quality. Some measures have been undertaken to improve water quality, including the installation of water filtration plants to reduce the turbidity and colour of the water and the use of chloramination and chlorination to control the bacterial content of the water. However, the water supplied to consumers has high levels of salinity. From July 1975 to June 1988, the salinity in the Murray at Murray Bridge varied from 140 mg/L to 820 mg/L, with an average of 400 mg/L (Dandy and Crawley, 1992). In comparison, the World Health Organisation's maximum desirable level for human consumption is 500 mg/L (WHO, 1984).

3.2.1 The Salinity Problem in the River Murray

The Murray-Darling river system has a major salinity problem, both in terms of river salinity and land salinisation. The River Murray provides the only 'natural' means for the removal of salts from the Murray basin. The salinity of the River Murray varies considerably along its length (Figure 3.5). In the upper reaches, river salinity is only a few times higher than that of rainwater. However, the salinity increases dramatically in the lower part of the river, and at Murray Bridge, the salinity is approximately 7 to 20 times greater than that at Jingellic on the upper Murray above Hume Dam (EWS, 1990). In the upstream states, the increase of salinity with distance is small. This is due to the large number of tributaries with low salinities and the absence of inflows of saline groundwaters. In South Australia, on the other hand, the increase of salinity with distance is very rapid, as there are extensive inflows of highly saline groundwater and there are no significant tributaries.

The salinity of a water sample is defined as the concentration of all the solids dissolved in it, and is hence dependent on the salt load as well as the volume of water, i.e. concentration (mg/L) = salt load (tonnes) / flow (GL). In most natural waters, the dissolved solids will be made up entirely of salts. The major constituents of the salinity of the River Murray are chlorides, bicarbonates and sulphates of calcium, sodium, magnesium and potassium (Mackay and Eastburn, 1990).

Salinity is measured in 'EC' units. 'EC' units are defined as micro Siemens / cm at 25° Celsius. 'EC' units measure the conductivity of saline water, which increases with increased concentrations of dissolved salts (EWS, 1990). The conductivity is also

dependent on the temperature and ionic composition of the water, and corrections have to be made to adjust the conductivity measurement to the standard temperature of 25° Celsius. For the River Murray, a typical factor used to convert 'EC' units to salinity as mg/L total dissolved solids (TDS) is 0.6, i.e. TDS (mg/L) = (0.6) EC units (Mackay and Eastburn, 1990).

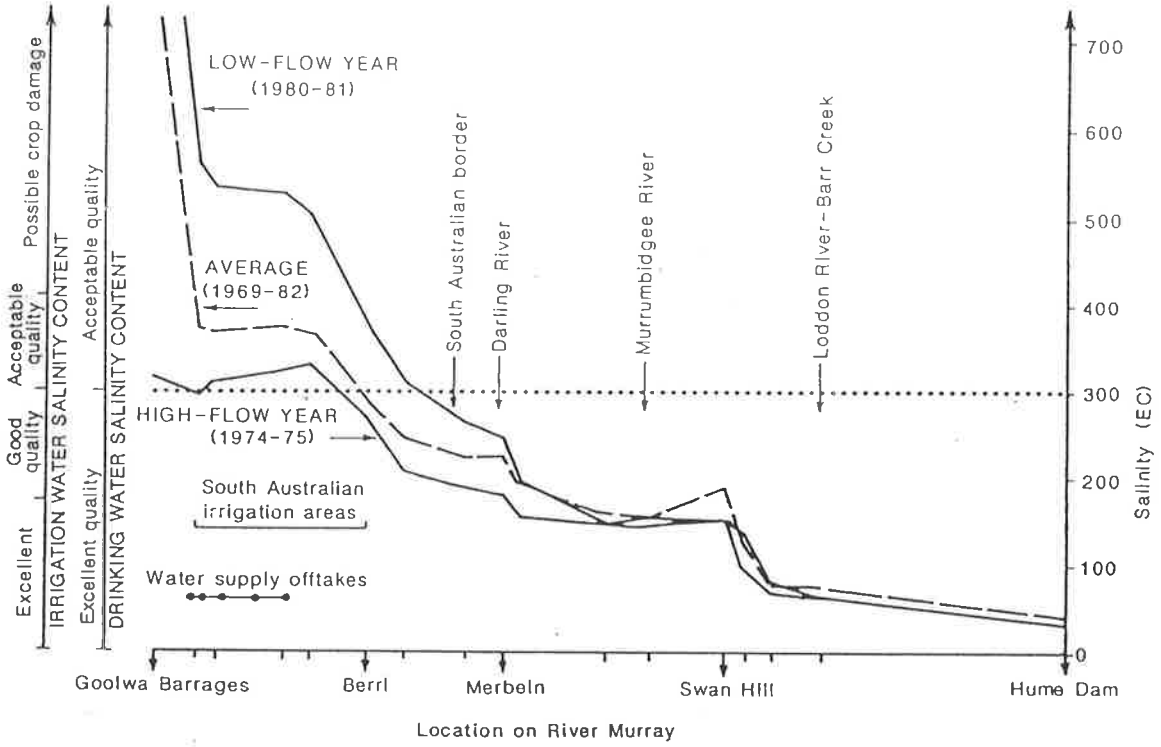


Figure 3.5: River Murray Salinity Profile (Source: Murray-Darling Ministerial Council, 1988)

3.2.1.1 Factors Causing Salinity

Sources of salt (Mackay and Eastburn, 1990)

Salt comes from the following sources:

1. Small quantities of salt are contained in rain.
2. Salt is deposited by wind.
3. Salts are released by the breakdown of rock minerals and the structural weathering of rocks which contain salts.

Processes by which salt enters the river

In the normal range of operational discharge (3,000 ML/day to 15,000 ML/day), salt enters the river in the following ways (Jakeman et al., 1986; EWS, 1990; Mackay and Eastburn, 1990):

1. Inflows of saline groundwater. The salinity of the groundwater is high as it dissolves salts from the soil and rock structure through which it moves and also contains salts that have been leached from the surface (Figure 3.6).
2. Saline inflows from drains. Drains recycle the salt present in irrigation water and often intercept saline groundwaters.
3. Inflows of saline tributaries.
4. Natural surface runoff, which carries with it salts that have been deposited on the land.
5. The weathering of mineral soil particles in river sediments.

The above processes are illustrated in Figure 3.7.

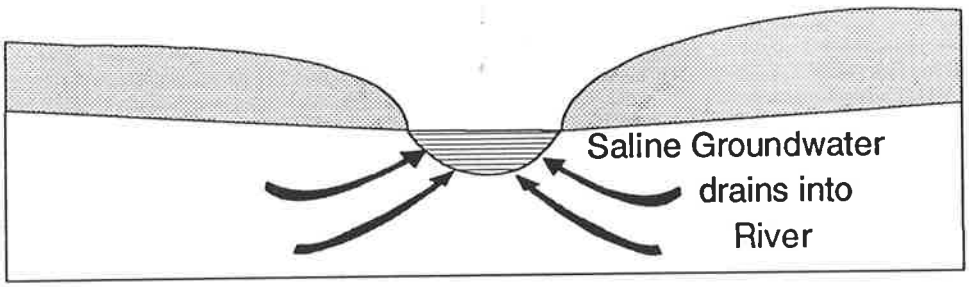


Figure 3.6: Inflows of Saline Groundwater (Source: Adapted from EWS, 1990)

Groundwater inflows are the largest single source of salt in the River Murray. Maunsell and Partners (1979) estimate that in a typical year, the total salt inflow to South Australia (measured at Lock 6) is approximately 1,100,000 tonnes, of which 600,000 tonnes are attributed to tributary inflows, 250,000 tonnes to drainage inflows and the remaining 250,000 tonnes to groundwater inflows. However, it is estimated that a further 500,000 tonnes of salt are added in South Australia; all due to groundwater inflows.

Flooding has a profound effect on the way salt enters the river. The effect of a flood hydrograph is to first suppress, then to accentuate groundwater inflow. While the discharge is high, saline accessions are attributed to the following phenomena (Jakeman et al., 1986; Mackay and Eastburn, 1990):

1. The contents of backwaters and anabranches are flushed into the river. Backwaters and anabranches are very saline, as they intercept saline groundwaters. The concentration of the salty water trapped in these systems is further increased as a result of evaporation.
2. Salt residues are leached from banks located above the normal operational range of river levels.
3. Turbulent flow increases bank erosion of saline soils and mixes saline waters trapped in deep holes in the riverbed.

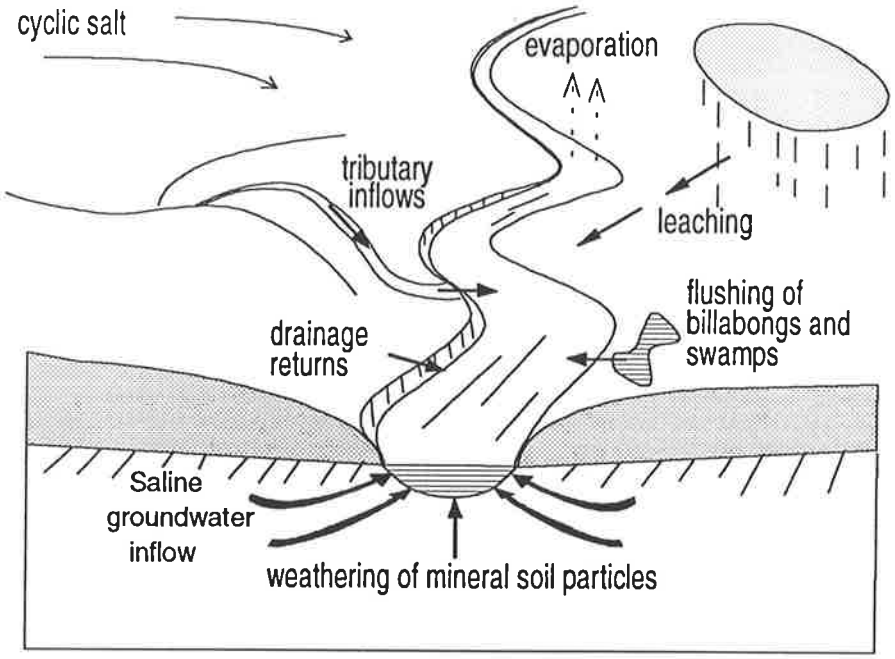


Figure 3.7: Natural Sources of Salinity (Adapted from Jakeman et al., 1986)

3.2.1.2 Factors Increasing Saline Accessions

Development within the catchment has increased saline accessions through the following mechanisms (Jakeman et al., 1986; Ghassemi et al., 1991; EWS, 1990):

1. Land clearance:
   Trees intercept rain and transpire. Consequently, only a small percentage of rainfall reaches the groundwater system. However, as a result of European settlement, the land has been cleared and a lot of deep-rooted trees have been removed, causing more precipitation to reach the groundwater. The effect of this is to mobilise soil salts and to raise the groundwater level resulting in more saline groundwater discharges into the river.

2. Irrigation drainage:
   European settlement also resulted in the onset of irrigation practices. Usually, excess irrigation water is applied in order to flush salts beyond the root zone. The excess water reaches the groundwater, forming a groundwater mound. As a result, groundwater inflows to the river are increased. In addition, waterlogging of rootzones and loss of agricultural productivity occurs if the groundwater level is high enough.

3. River structures:
   River structures, such as locks and weirs, increase river levels by several metres. The resulting hydrostatic pressure forces saline groundwaters into the river on the downstream side (Figure 3.8).

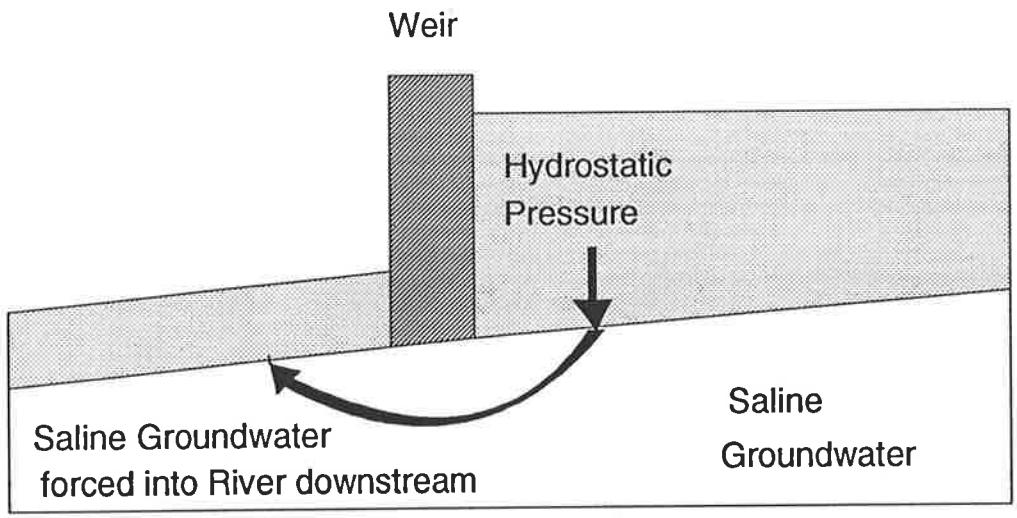


Figure 3.8: Salt Inflows Aggravated by River Structures (Source: Adapted from EWS, 1990)

River salinity is also affected by other physical processes, such as evaporation and rainfall:

- Evaporation causes an increase in the concentration of salts.
- During high rainfall periods, more salt is washed into the river, but the salinity is less due to the increased flow.

3.2.1.3 Effects of High Salinity

In general, the more saline the water, the fewer the uses to which it can be put. High salinity levels have two interconnected effects (Mackay and Eastburn, 1990):

1. They affect users of River Murray water.
2. They cause salinisation of soils if groundwater levels are sufficiently high.

Effects on domestic users

High salinities of river water have the following effects on domestic users (EWS, 1990; Mackay and Eastburn, 1990):

1. Palatability:
   When salinity levels are high, the taste of the water becomes noticeable.

2. Hardness:
   As the proportion of magnesium and calcium ions in solution increases, so does the hardness of the water. As a result, more soap and detergents are required, water softeners may be needed, stunting of growth of domestic plants may occur, scaling occurs in pipes and heaters and fabric wear is increased.

3. Corrosion:
   Saline water increases the rate at which pipes and fittings corrode and hence results in higher maintenance and replacement costs.

The effects described above carry with them economic costs. Dwyer Leslie Pty. Ltd. (1984) estimate the cost for domestic users in Adelaide as 35.9 cents per year per mg/L of salinity per household (in 1983 Australian cents). This would result in a total cost to domestic users in Adelaide of approximately $118,500 per year per mg/L (in 1983 Australian dollars). Mackay and Eastburn (1990) provide a more recent estimate for the cost to domestic users, which is equal to 25 cents per household per EC unit per year.

Effects on industrial users

Saline water used for industrial purposes causes corrosion, scaling and poor steam quality. This results in increased energy and chemical treatment costs. There are major costs involved in the steam generation process (Australian Mineral Development Laboratories (AMDEL), 1983), resulting from:

- Heat energy lost due to blow down.
- Water lost due to blow down.
- Chemicals lost due to blow down.
- Softening of boiler feed water.

- Demineralisation of boiler feed water.

AMDEL (1983) estimate the cost to industrial users of saline water as $5600 per year per mg/L (in 1983 Australian dollars) for salinities in the range of 300 mg/L to 500 mg/L. Mackay and Eastburn (1990) estimate the cost to industrial users of River Murray water as $6500 per EC unit per year.

Effects on agricultural users

Irrigators along the River Murray rely entirely on river water. However, irrigation with saline water, combined with the inefficient use of water, can lead to soil salinisation. Saline water reduces the permeability of clay soils, making surface soils sticky and waterlogged when wet and hard when dry. Soil salinisation is usually only a problem in arid and semi-arid regions, as rainfall leaches salts through the soil in non-arid regions, preventing soil salinisation. Buringh (1979) estimates that soil salinisation causes the worldwide loss of arable land at a rate of 3 ha/min (i.e. 1.6 Mha per year) and Kovda (1983) estimates the same value to be 1 to 1.5 Mha per year. Soil salinisation has a number of negative effects on the environment, the economy and on society (Ghassemi et al., 1991; Mackay and Eastburn, 1990), including:

1. It reduces soil productivity and may transform fertile soils with high yields into barren land.
2. It reduces crop yield, limits the choice of crops that can be grown and may kill vegetation (Figure 3.9).
3. It may lead to loss of habitat and reduced biodiversity.
4. It can cause the dislocation of farm populations.

Saline irrigation water also has the following direct effects on crops (Mackay and Eastburn, 1990):

1. Osmotic effects:
   The roots of plants have the ability to exclude salt from water they absorb. However, this becomes more difficult as the salinity of the irrigation water increases. As a result, the plants suffer from water stress.

2. Toxic effects:
   At high salinity levels, ions enter the root system and can build up to toxic levels in the roots and leaves.

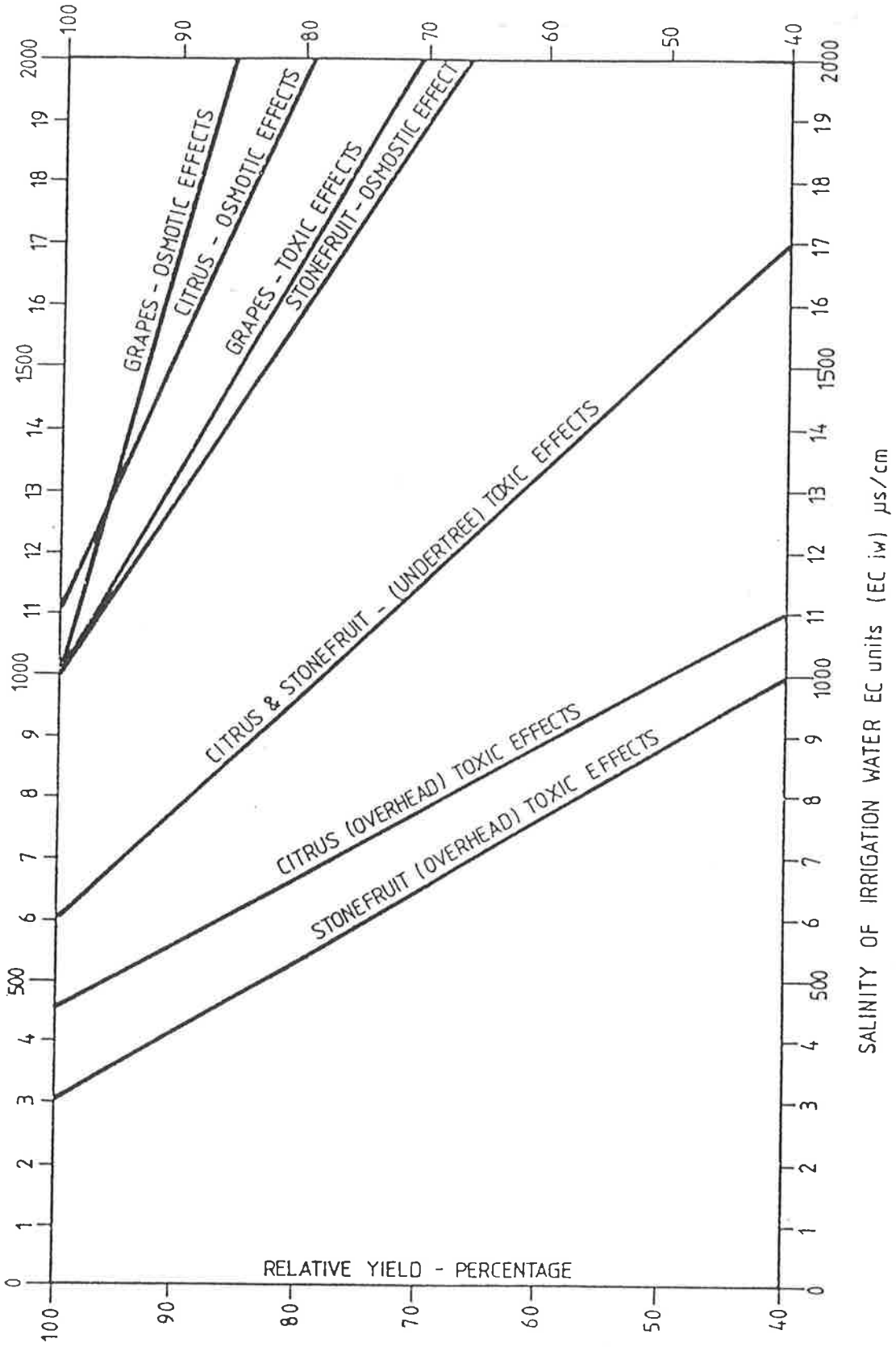


Figure 3.9: The Effect of Salinity on Crop Yield (Source: Jakeman et al., 1986)

The economic costs of salinity to irrigated agriculture is a combination of production losses and corrosion damage to water supply systems.

Total costs

The EWS (1990) estimates the total costs for domestic, industrial and agricultural users per change of one EC unit in the average salinity at Morgan to be $82,000 per year (in 1988 Australian dollars). Mackay and Eastburn (1990) put the same figure at $70,000.

3.2.1.4 Remedial and Preventive Measures

Remedial and preventive measures to combat the salinity problem include (Ghassemi et al., 1991; EWS, 1990; Mackay and Eastburn, 1990; Dandy and Crawley, 1992):

1. Reducing saline accessions by:
   - Improving surface drainage.
   - Improving irrigation techniques.
   - Lining irrigation channels.
   - Land forming.
   - Establishing deep-rooted vegetation.
   - Intercepting saline groundwater.
   - Intercepting saline drainage.
   - Improving water distribution systems.

2. Lowering water tables by:
   - Groundwater pumping.
   - Tile drainage.

3. Releasing dilution flows from reservoirs.

4. Using surface water in conjunction with groundwater resources.

5. Water pricing to reduce water consumption.

6. Optimising pumping from the River Murray with respect to water quality.

The choice of remedial and preventive measures depends on the circumstances and usually a combination of the above measures is adopted.

3.2.2 Case Study

The case study considered in this chapter is concerned with forecasting salinity in the River Murray. Salinity forecasts at important locations are needed for daily management purposes as well as planning and analysis purposes. Salinity forecasting models (Jakeman et al., 1986; Thomas and Jakeman, 1983) can be used to:

1. Predict downstream salinity concentrations from upstream data to assist in management strategies, such as salinity dilution by the release of additional discharges from upstream storages.
2. Assess management strategies aimed at reducing the influx of aquifer salinity into the river.
3. Provide accurate advice to irrigators.
4. Model the "pre-scheme" and "post-scheme" salt regimes for salinity reduction schemes in order to measure their effectiveness.
5. Develop optimal pumping and/or flow release strategies in order to achieve desired salt concentrations.
6. Adjust operations to cater for contigencies such as salt slugs or periods of high salinity.

In this case study, Murray Bridge (Figure 3.4) was chosen as the site at which salinity forecasts will be obtained. The offtake from the River Murray at Murray Bridge forms a significant part of Adelaide's water supply, which is pumped to Adelaide via the Murray Bridge to Onkaparinga pipeline. By knowing salinity values in advance, adequate pumping policies can be developed such that more water can be supplied at times of low salinity and pumping at times of high salinity can be reduced. In fact, Dandy and Crawley (1992) have developed an optimisation model for obtaining an optimum pumping policy taking salinity into account. Assuming that salinities can be forecast perfectly several weeks in advance, the model indicates that a reduction of 45 mg/L can be achieved in the average salinity of the water supplied. The increased pumping cost to achieve this would be $2.0 Million per year. However, the reduced salinity is estimated to reduce the salt damage costs by $6.0 Million per year, resulting in a benefit/cost ratio of 3.0.

3.3 Salinity Forecasting Models

3.3.1 Forecasting Methods

Possible methods for forecasting salinity in the River Murray at Murray Bridge include:

- ANN methods.
- "Traditional" time series methods.
- Physically based methods using the St. Venant equations and the one-dimensional convective diffusion equation (James, 1984).

As discussed in Sections 2.1.9.5 and 2.3, ANNs are suitable for forecasting water resources time series and provide an alternative to more traditional time series methods. ANN methods were considered suitable for this case study, as the following criteria, (exploiting the strength of ANN methods (Section 2.1.9)), were met:

- A considerable amount of input data is available.
- Non-linear relationships are suspected.
- There is considerable noise (measurement and sampling) in the data collected (Thomas and Jakeman, 1983).
- It is difficult to prescribe the exact mathematical relationship between the variables (Thomas and Jakeman, 1983).

Back-propagation networks were used in this case study, as they are well suited to, and have been successfully used for, forecasting applications. Although more complex networks exist (e.g. recurrent networks), back-propagation networks generally give good performance and take less time to train (Chakraborty et al., 1992).

It was also decided to consider time series models of the ARMA type, as these have been traditionally used to model water resources time series (Section 2.2.13.12) and provide a good basis of comparison for the ANN model. ARIMA type models were developed in preference to more advanced time series models, such as time variable parameter (TVP) models (Section 2.2.13.10), as they have been in common usage since at least 1970 and are among those forecasting models most successfully applied in practice (Tang et al., 1991).

Physically based models mathematically simulate the physical processes underlying the relationships of interest (Hipel, 1985). Some problems associated with physically based models include that they are very complex, incorporate a large number of parameters in order to describe physical phenomena, all of which must be calibrated (Tong et al.,

1985) and are generally only an approximation to reality due to the great complexity of natural systems (Hipel, 1985).

In this research, a physically based model was considered to be too expensive and difficult to develop for the following reasons:

- It would be very difficult to accurately model all the processes affecting river salinity described in Section 3.2.1.1.
- Additional data would need to be collected to enable the model to be calibrated and verified.
- Sufficiently comprehensive, controlled experiments to determine the precise relationships between the variables cannot be carried out (Thomas and Jakeman, 1983).
- Coincident flow and salinity measurements are required at the prediction location (Thomas et al., 1986). These are not available at Murray Bridge.

Consequently, it was decided to develop a univariate ANN model (Section 3.4), a multivariate ANN model (Section 3.5), a univariate time series (ARIMA) model (Section 3.6) and a multivariate time series (VARIMA) model (Section 3.7). The univariate models were developed, as they provide a good yardstick against which the performance of the more complex, multivariate models can be compared (Harvey, 1989).

3.3.2 Forecasting Period

Initially, a forecasting period of 14 days was selected, as this enables short term adjustments to be made to the pumping schedule. One and five day forecasts were also obtained, in order to assess the performance of the various models at different forecasting periods. Shorter term forecasts are also useful in providing advice to irrigators. The multivariate ANN model was also used to obtain a 28 day forecast, as this type of model proved to be the most successful for longer term forecasting. Forecasts of this length are required for the development of pumping schedules taking salinity considerations into account.

3.3.3 Available Data

ANN and time series models do not directly model the individual processes responsible for high levels of salinity. However, it is important to have a good understanding of the underlying physical processes affecting salinity, so that an informed choice can be made

about which input variables to include in the model, and to gain a good understanding of the limitations imposed on the model by the available data.

The main processes affecting salinity in the River Murray include the transport of salt from upstream locations and saline accessions along the length of the river (Thomas and Jakeman, 1983). In the South Australian reaches of the Murray, accessions are almost entirely due to the inflow of saline groundwater.

The factors affecting salinity transport include flow and upstream salinities, whereas accessions of saline groundwater are a function of flow as well as river and groundwater levels (Jakeman et al., 1986). Consequently, upstream salinities, flow, river levels and groundwater levels were identified as the dominant variables. All of these variables, with the exception of groundwater levels, are measured on a daily basis at various locations along the Murray. However, groundwater levels change slowly with time (Jakeman et al., 1986) and thus do not have a significant influence on the temporal variation of groundwater accessions. As a result, the available data should account for salinity transport and groundwater accessions. However, no information is available on other accessions, including saline inflows from drains, natural surface runoff and the flushing of anabranches and backwaters.

The data used were supplied by the Engineering and Water Supply Department (EWS) of South Australia and are summarised in Table 3.2 (refer to Figure 3.10 for locations). All data were available from 01-12-1986 to 30-06-1992. The abbreviations shown in Table 3.2 will be used in the remainder of this chapter and are also shown in Appendix B.

Table 3.2: Available Data (Daily)

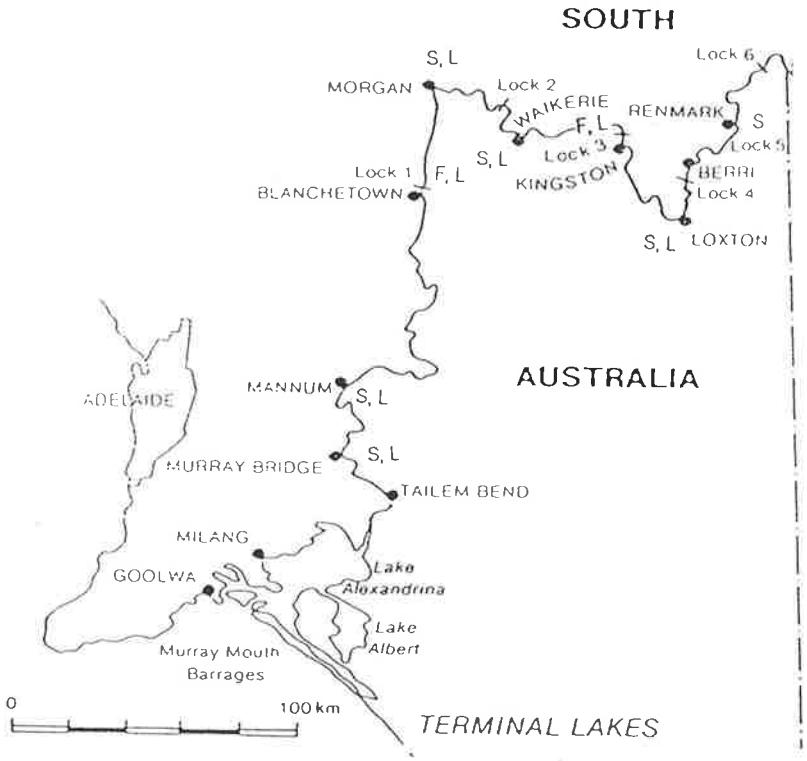
Location	Data Type	Abbreviation
Murray Bridge	Salinity	SMB
Mannum	Salinity	SMN
Morgan	Salinity	SMO
Waikerie	Salinity	SWE
Loxton	Salinity	SLO
Lock 5 Upper	Salinity	SL5U
Lock 1 Lower	Flow	FL1L
Overland Corner	Flow	FOC
Murray Bridge	River Level	LMB
Mannum	River Level	LMN
Lock 1 Lower	River Level	LL1L
Lock 1 Upper	River Level	LL1U
Morgan	River Level	LMO
Waikerie	River Level	LWE
Overland Corner	River Level	LOC
Loxton	River Level	LLO

3.3.3.1 Inspection of Plots of Data

Prior to detailed data analysis, it is important to inspect the plots of the time series listed in Table 3.2 in order to get a visual indication of any seasonal effects, trends, changes in variance and temporal and spatial relationships between them. Inspection of the plots also enables the detection of outliers, discontinuities and turning points.

Salinity Data:

A plot of the time series of salinity at Murray Bridge (SMB) from 01-01-1987 to 30-06-1992 is shown in Figure 3.11. It can be seen that there is a definite seasonal variation, with high salinities in the first half of the year and low salinities in the second half. However, the pattern is not regular and the start, end and duration of the periods of high and low salinity vary from year to year. The data exhibit no apparent trend. Over the five and a half year period considered, salinity ranged from 261 to 1116 EC units, with a mean of 599 EC units and a standard deviation of 199 EC units.



L = River Level Data, S = Salinity Data, F = Flow Data

Figure 3.10: Locations of Available Data

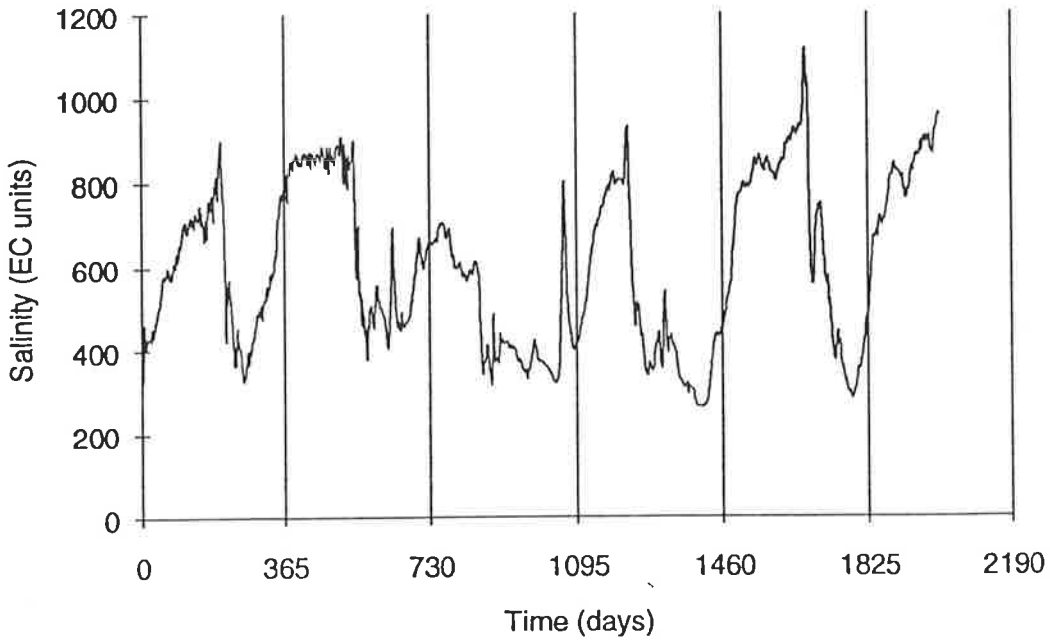


Figure 3.11: Salinity at Murray Bridge (SMB) (1987 to 1992)

The means, standard deviations, maxima and minima of the salinity time series upstream of Murray Bridge are shown in Table 3.3. As expected (Section 3.2.1), the mean, standard deviation and maximum values decrease with increasing distance upstream.

Table 3.3: Statistics of Salinity Time Series at, and Upstream of, Murray Bridge (1987 to 1992)

Time Series	Average (EC units)	Standard Deviation (EC units)	Maximum (EC units)	Minimum (EC units)
SMB	599	199	1116	261
SMN	580	187	1075	253
SMO	579	191	1061	177
SWE	571	180	1021	247
SLO	502	132	907	225
SL5U	417	109	712	179

Annual plots of salinity at Murray Bridge and salinity at locations upstream of Murray Bridge are shown in Figures 3.12 to 3.41. It can be seen that the shape of the salinity time series at upstream locations is very similar to that of salinity at Murray Bridge. As expected, the plots also show that the further upstream the site is, the lower the salinity values and the greater the lag between the upstream time series and the time series at Murray Bridge. The plots also show that:

1. There appear to be measurement errors at the following locations and times, as indicated by sharp drops or peaks in salinity which are confined to the location in question (i.e. which do not appear upstream or downstream of the site):
   - Mannum, 1989, days 64 and 100.
   - Morgan, 1989, day 116.
   - Morgan, 1990, day 305.

2. There appear to be saline accessions between Loxton and Waikerie, as a number of sharp peaks appear (or are accentuated) on the plot of salinity at Waikerie, which are not present (or are significantly diminished) on the plot of salinity at Loxton. This phenomenon usually occurs after the passing of the flood hydrograph and might be the result of the flushing of a billabong or an anabranch of the river. Examples can be seen by inspecting the plot of salinity at Loxton and the corresponding plot of salinity at Waikerie at the following times:

- 1988, days 250 to 300.
- 1989, days 125 to 175 and 300 to 350.

3. In some instances, salinities at upstream locations are higher than those at Murray Bridge, even when the lags due to salinity travel times have been taken into account. This seems anomalous, as salinity may be considered to be conservative and salinity at downstream locations generally increases as a result of saline accessions. Examples of the above phenomenon occur at the following locations and times:
   - Morgan, 1987, days 300 to 350.
   - Waikerie, 1987, days 300 to 350.
   - Morgan, 1989, day 116.
   - Waikerie, 1989, day 116.
   - Morgan, 1991, days 320 to 365.
   - Waikerie, 1991, days 320 to 365.
   - Loxton, 1991, days 320 to 365.

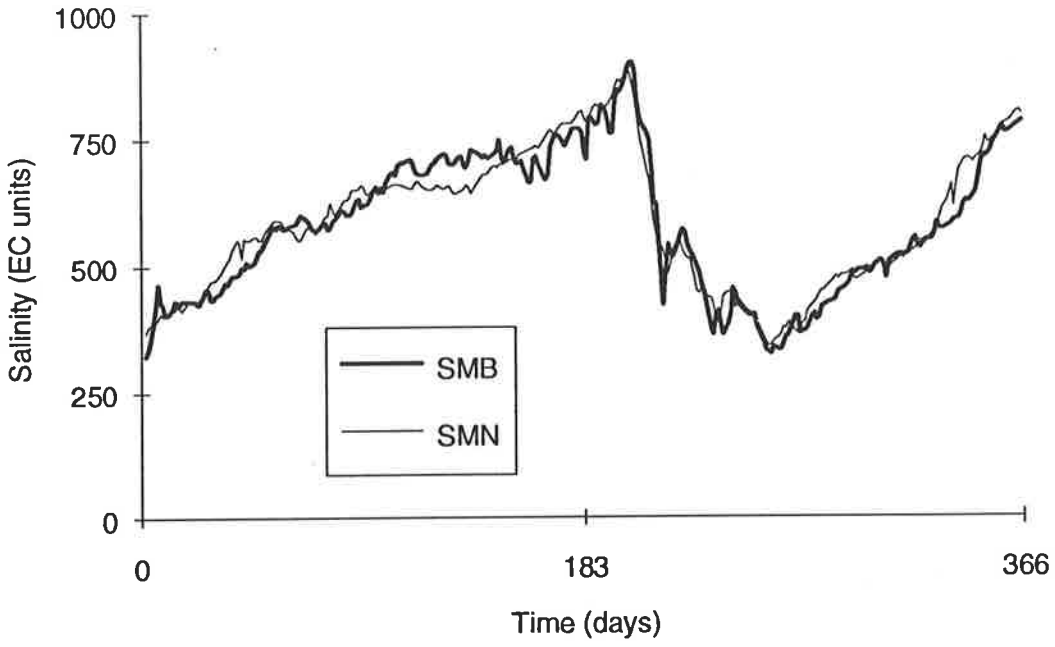


Figure 3.12: Salinity at Murray Bridge (SMB) and Salinity at Mannum (SMN) - 1987

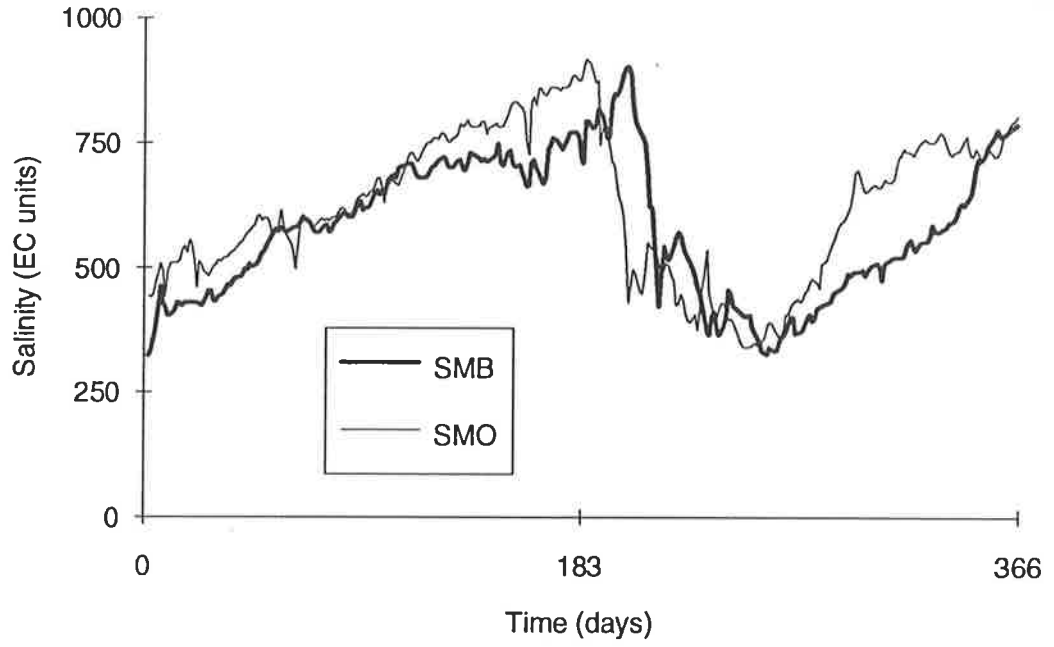


Figure 3.13: Salinity at Murray Bridge (SMB) and Salinity at Morgan (SMO) - 1987

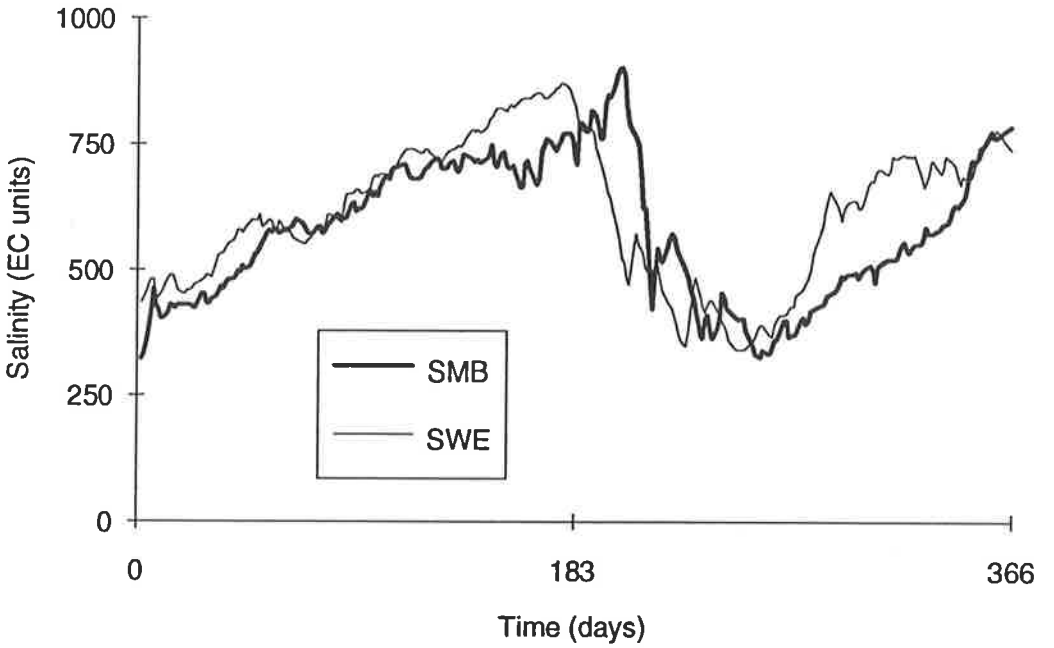


Figure 3.14: Salinity at Murray Bridge (SMB) and Salinity at Waikerie (SWE) - 1987

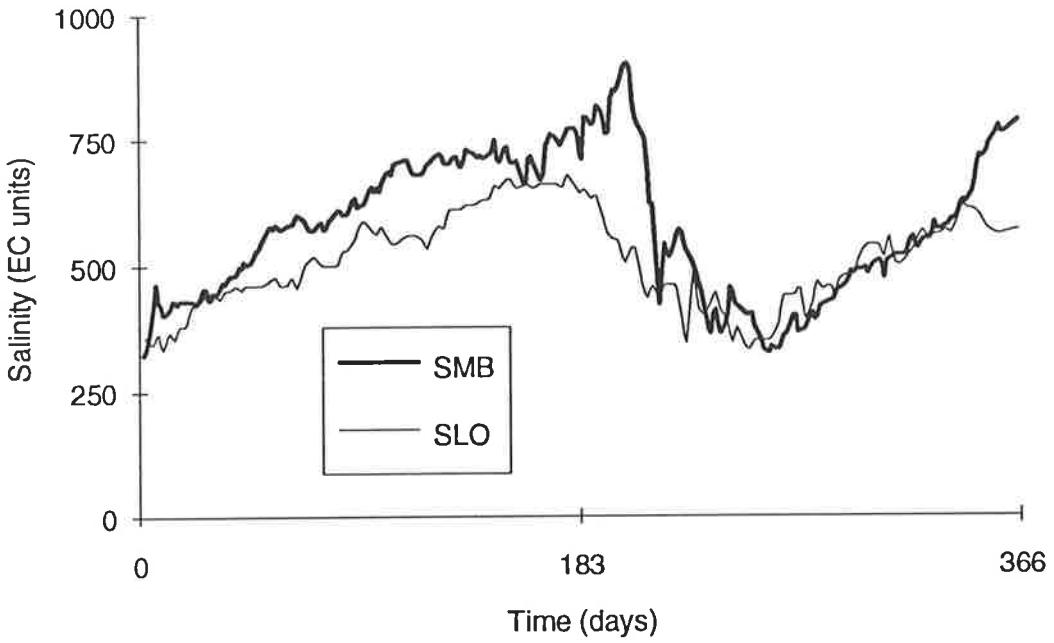


Figure 3.15: Salinity at Murray Bridge (SMB) and Salinity at Loxton (SLO) - 1987

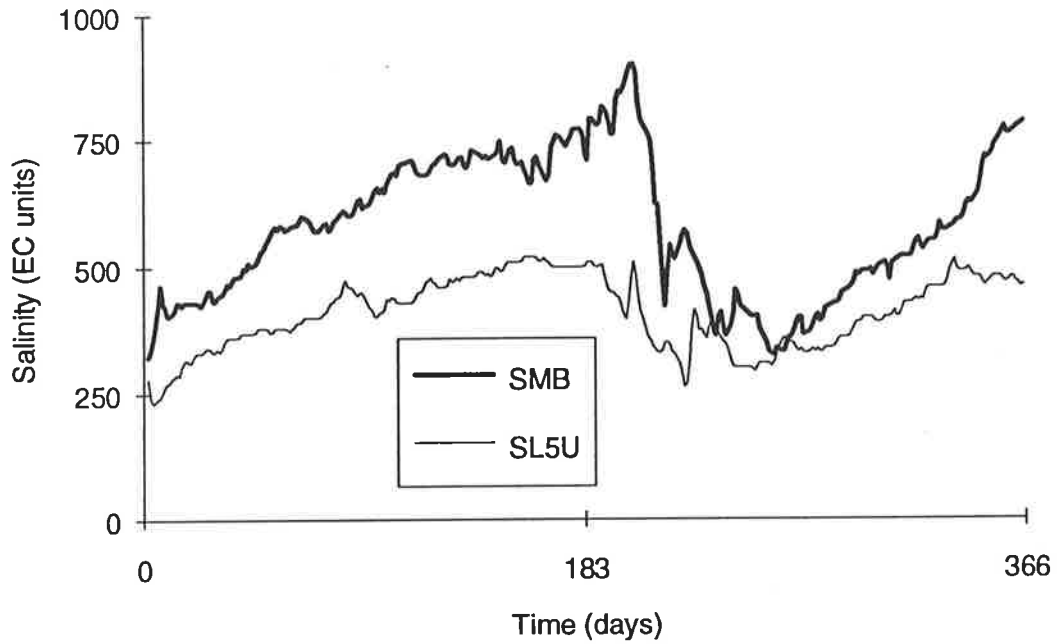


Figure 3.16: Salinity at Murray Bridge (SMB) and Salinity at Lock 5 Upper (SL5U) - 1987

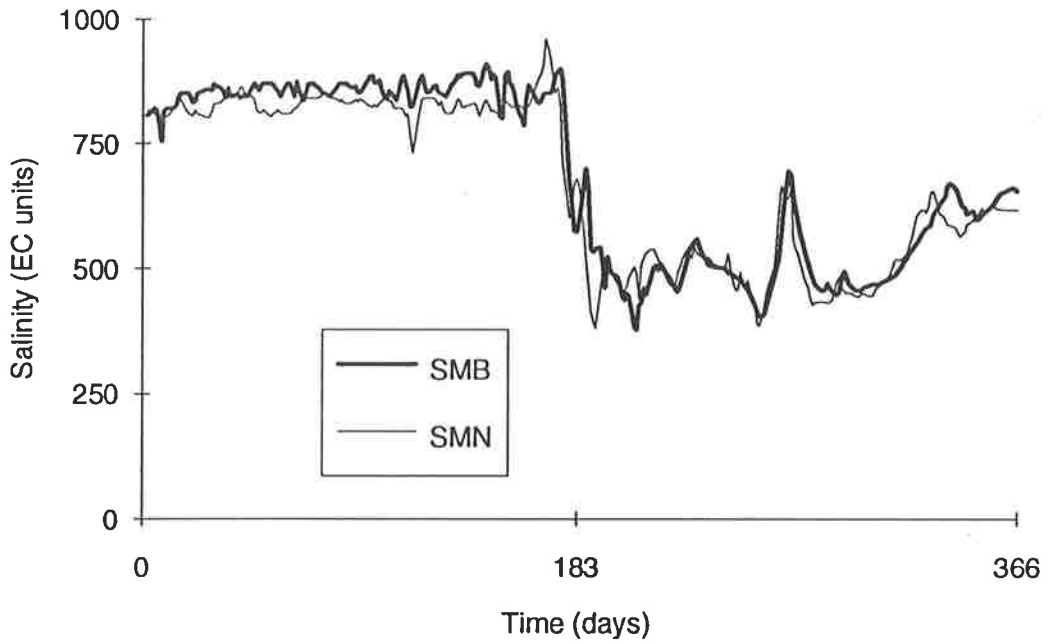


Figure 3.17: Salinity at Murray Bridge (SMB) and Salinity at Mannum (SMN) - 1988

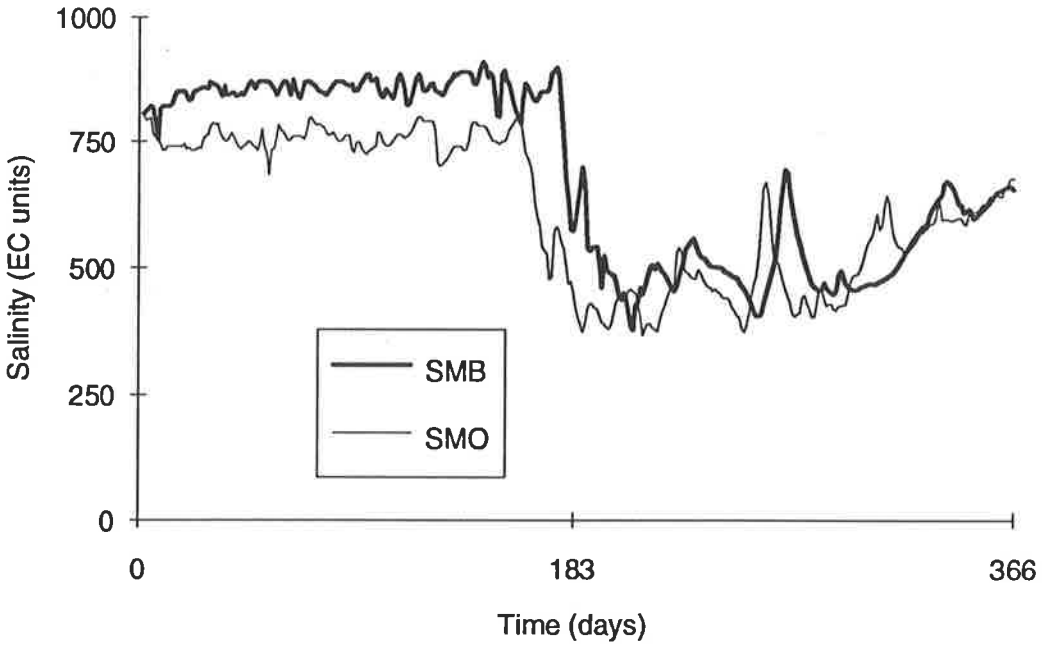


Figure 3.18: Salinity at Murray Bridge (SMB) and Salinity at Morgan (SMO) - 1988

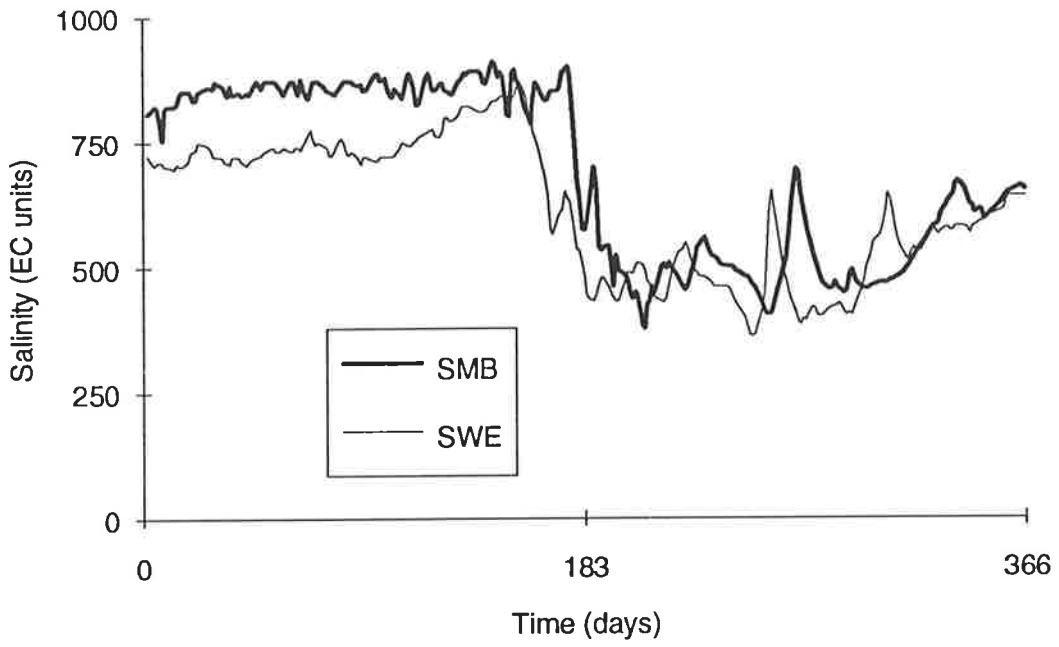


Figure 3.19: Salinity at Murray Bridge (SMB) and Salinity at Waikerie (SWE) - 1988

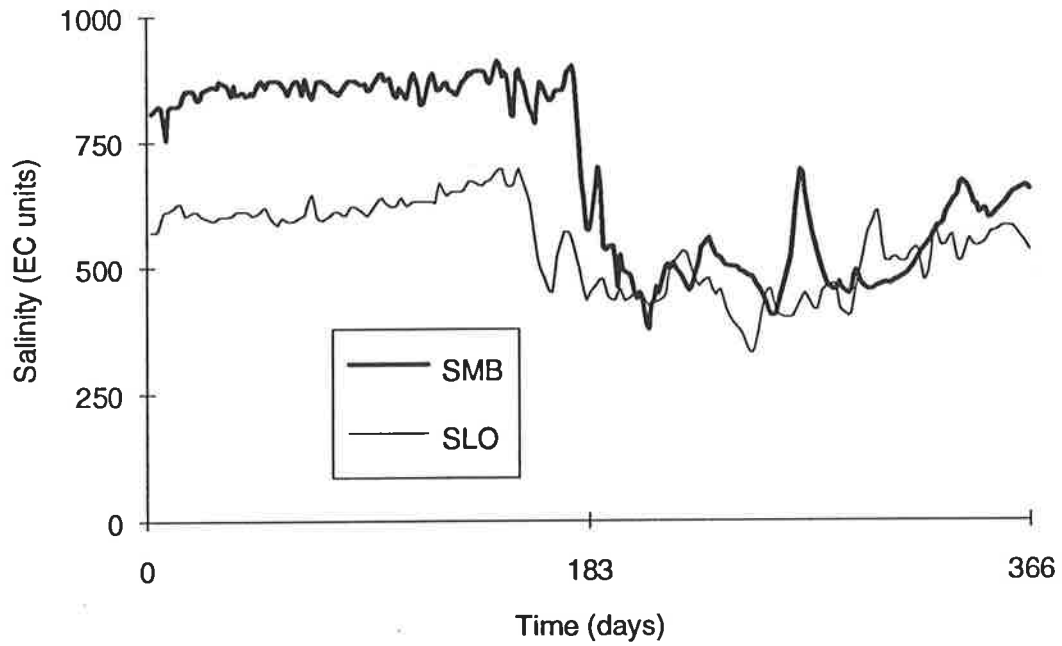


Figure 3.20: Salinity at Murray Bridge (SMB) and Salinity at Loxton (SLO) - 1988

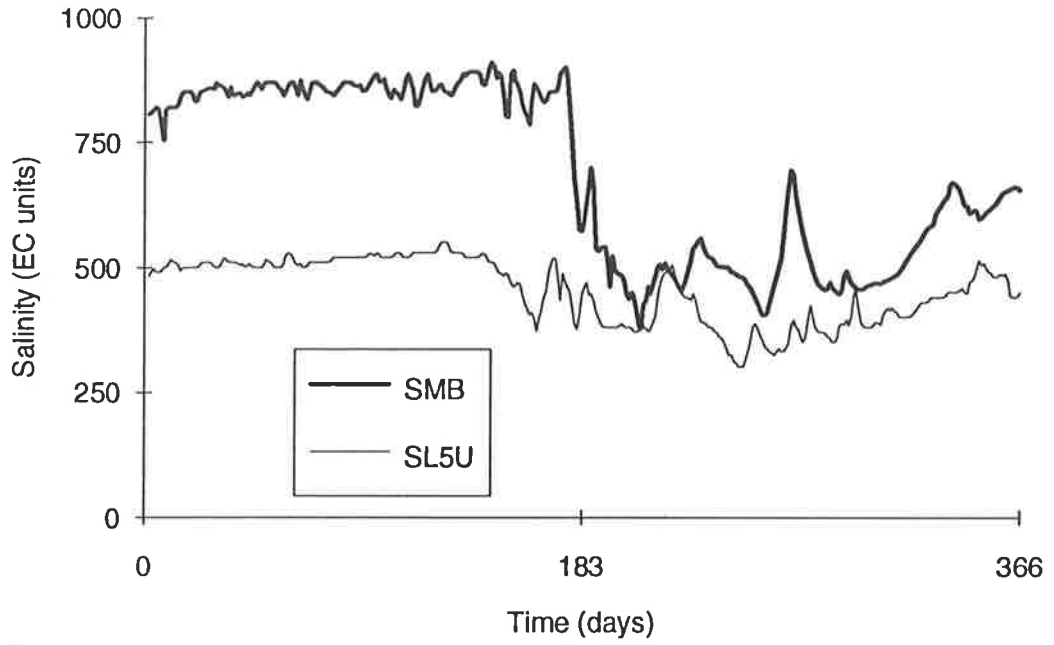


Figure 3.21: Salinity at Murray Bridge (SMB) and Salinity at Lock 5 Upper (SL5U) - 1988

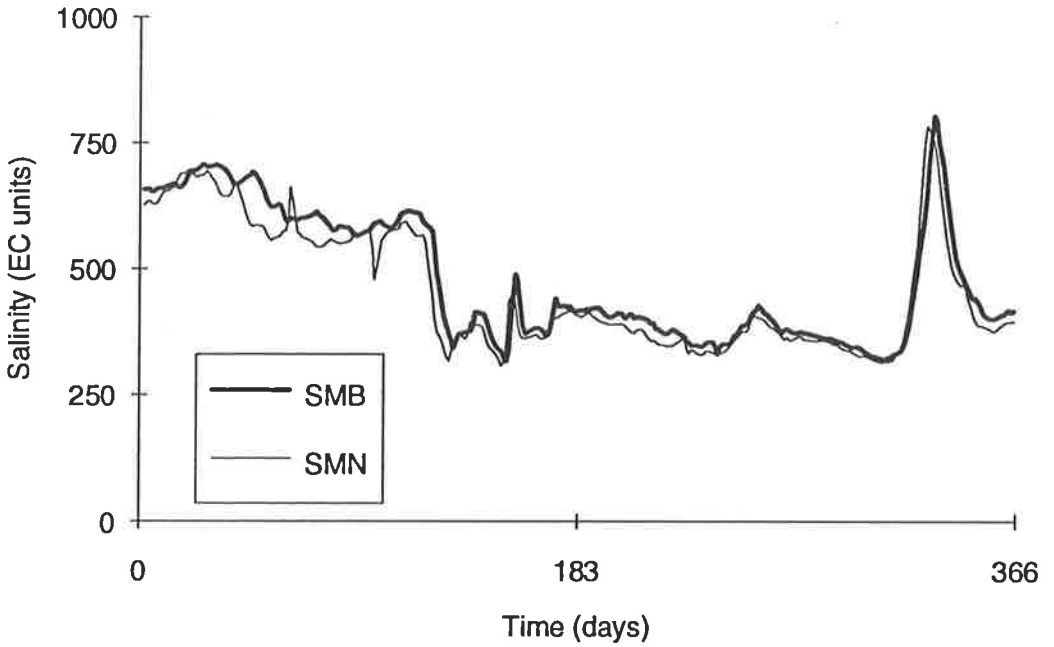


Figure 3.22: Salinity at Murray Bridge (SMB) and Salinity at Mannum (SMN) - 1989

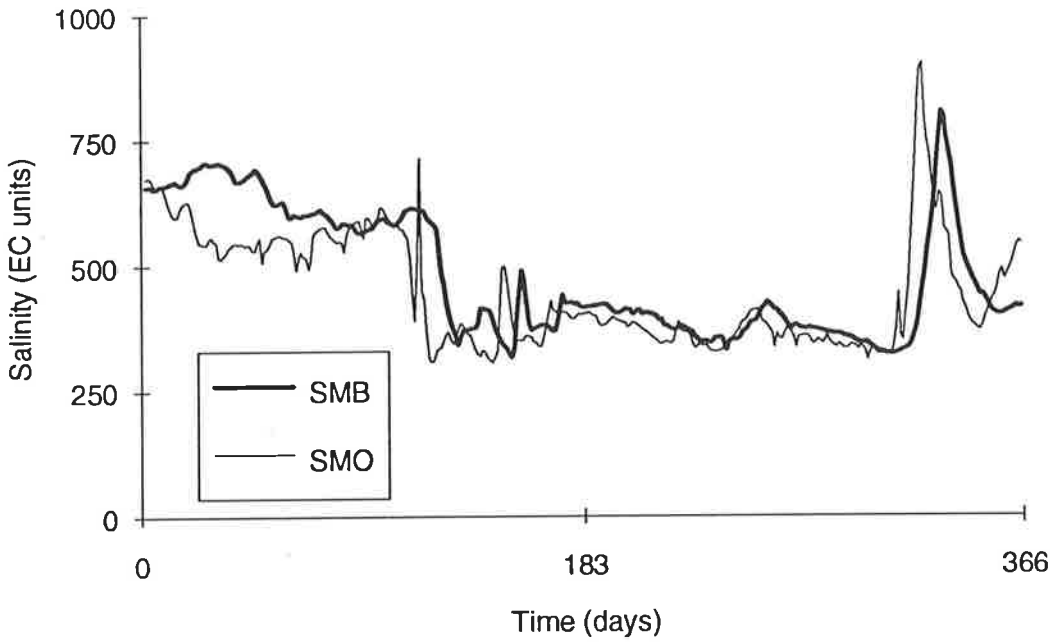


Figure 3.23: Salinity at Murray Bridge (SMB) and Salinity at Morgan (SMO) - 1989

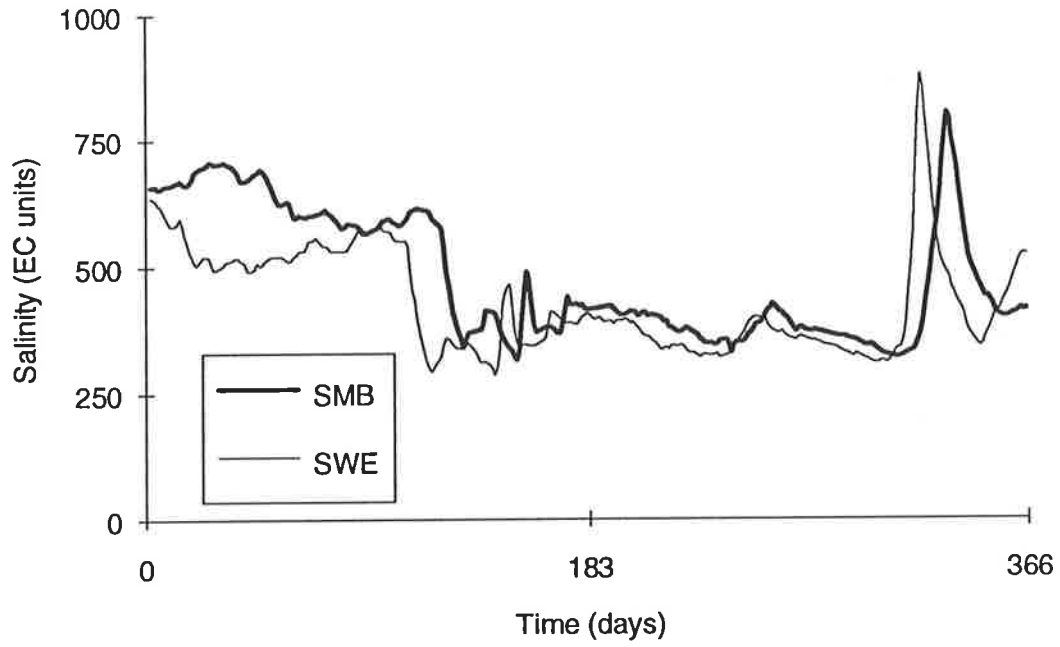


Figure 3.24: Salinity at Murray Bridge (SMB) and Salinity at Waikerie (SWE) - 1989

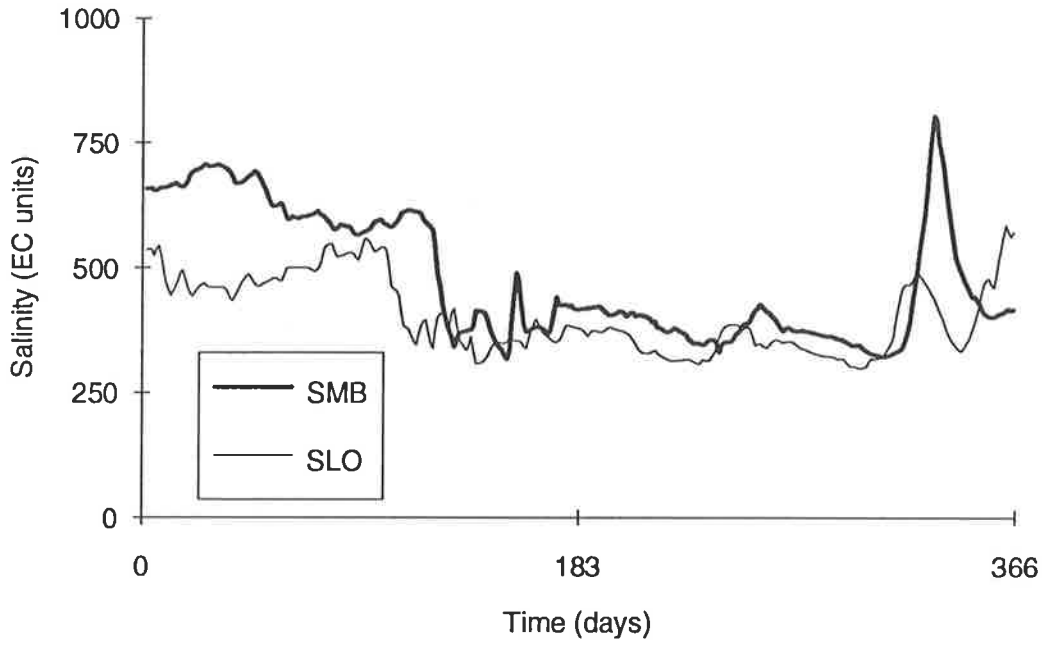


Figure 3.25: Salinity at Murray Bridge (SMB) and Salinity at Loxton (SLO) - 1989

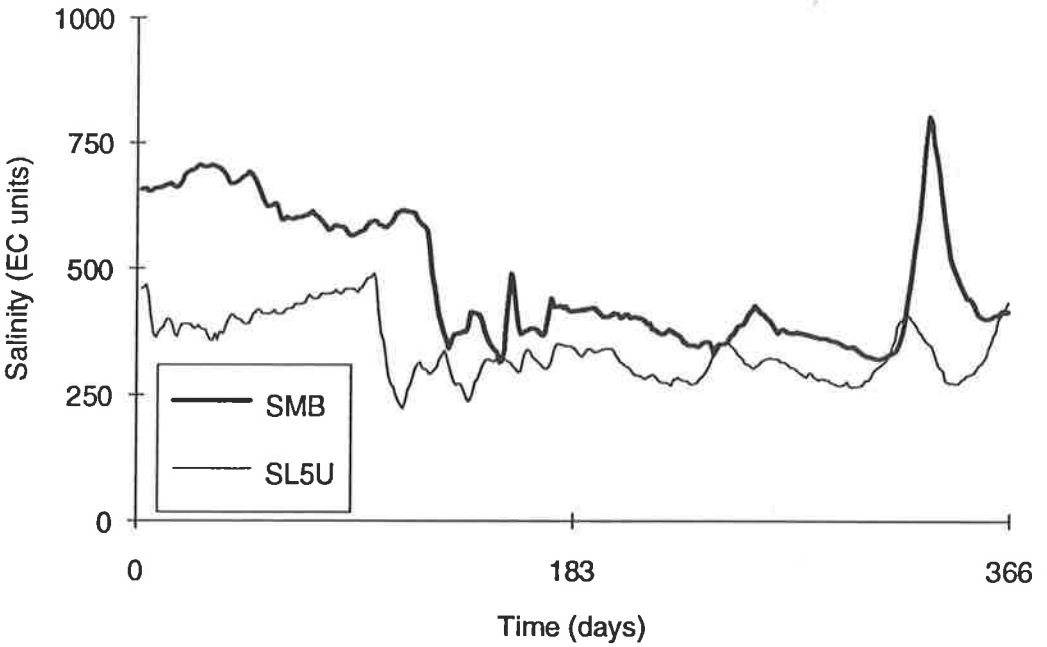


Figure 3.26: Salinity at Murray Bridge (SMB) and Salinity at Lock 5 Upper (SL5U) - 1989

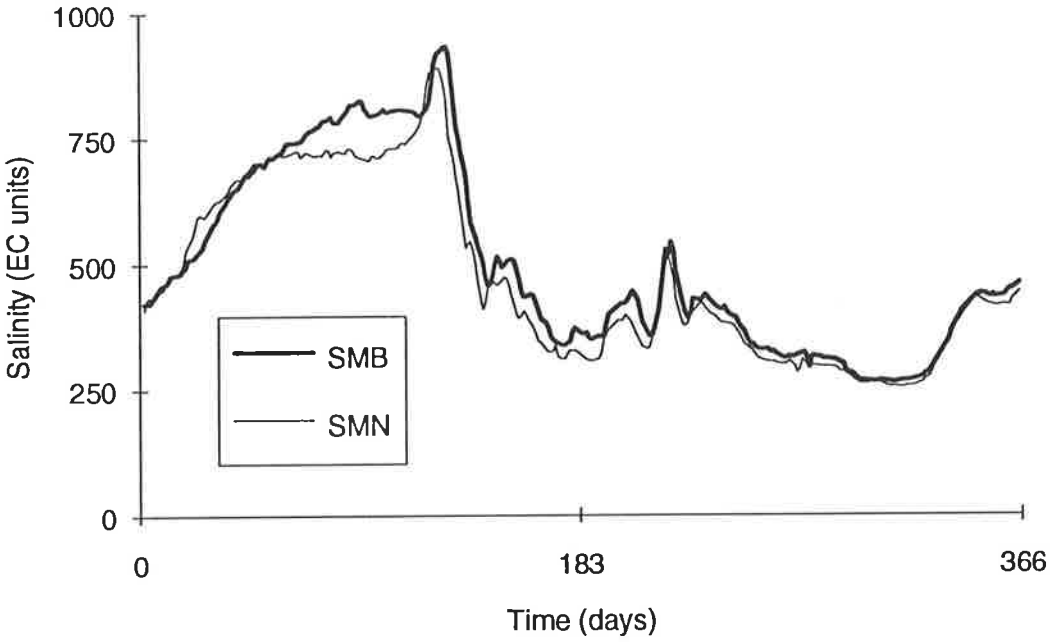


Figure 3.27: Salinity at Murray Bridge (SMB) and Salinity at Mannum (SMN) - 1990

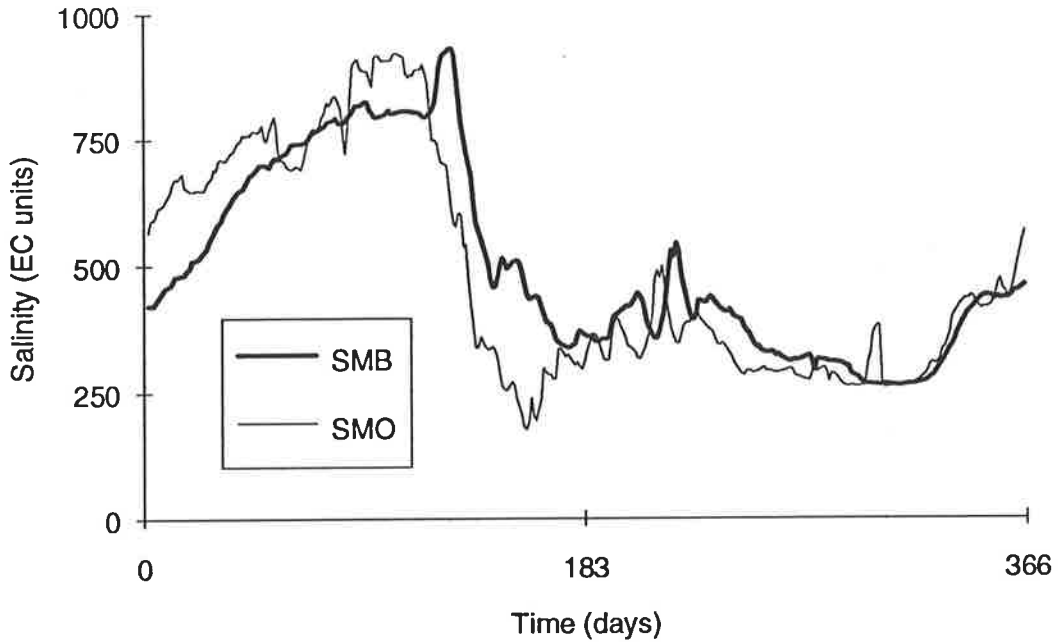


Figure 3.28: Salinity at Murray Bridge (SMB) and Salinity at Morgan (SMO) - 1990

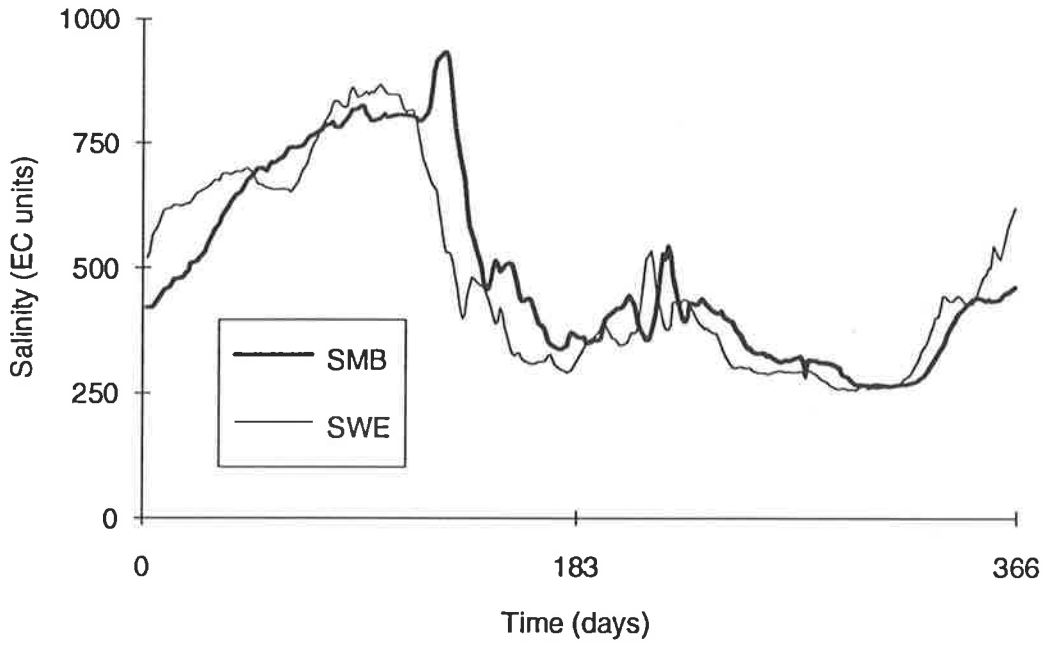


Figure 3.29: Salinity at Murray Bridge (SMB) and Salinity at Waikerie (SWE) - 1990

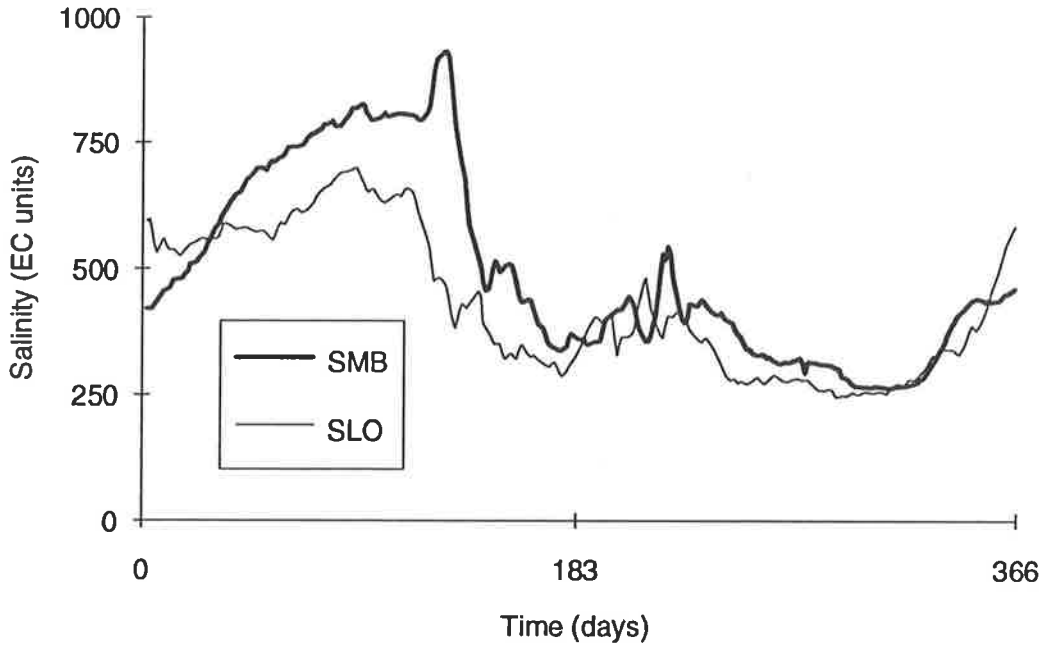


Figure 3.30: Salinity at Murray Bridge (SMB) and Salinity at Loxton (SLO) - 1990

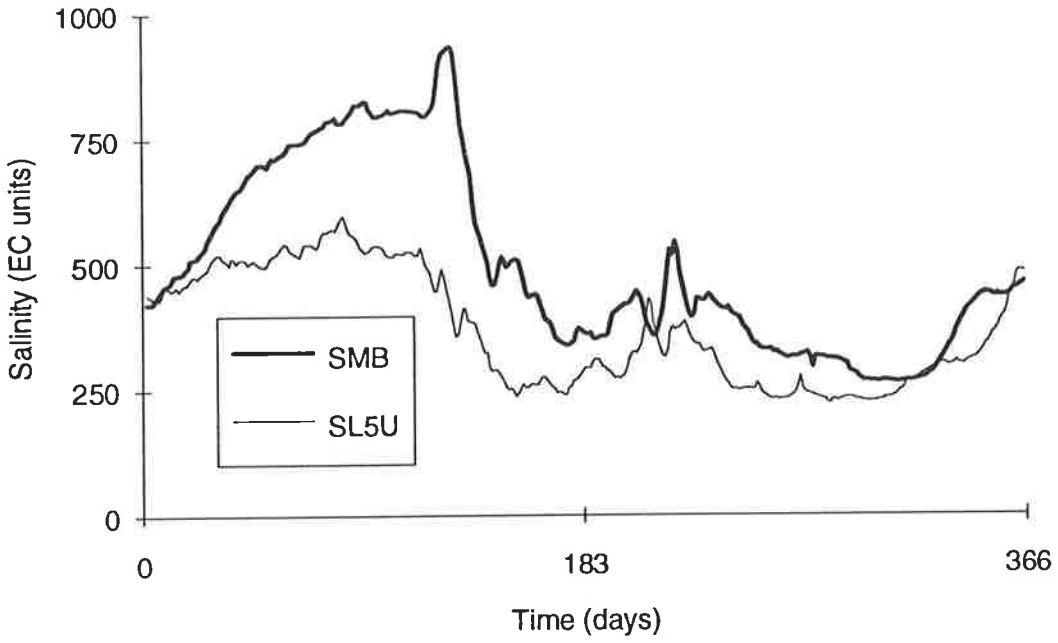


Figure 3.31: Salinity at Murray Bridge (SMB) and Salinity at Lock 5 Upper (SL5U) - 1990

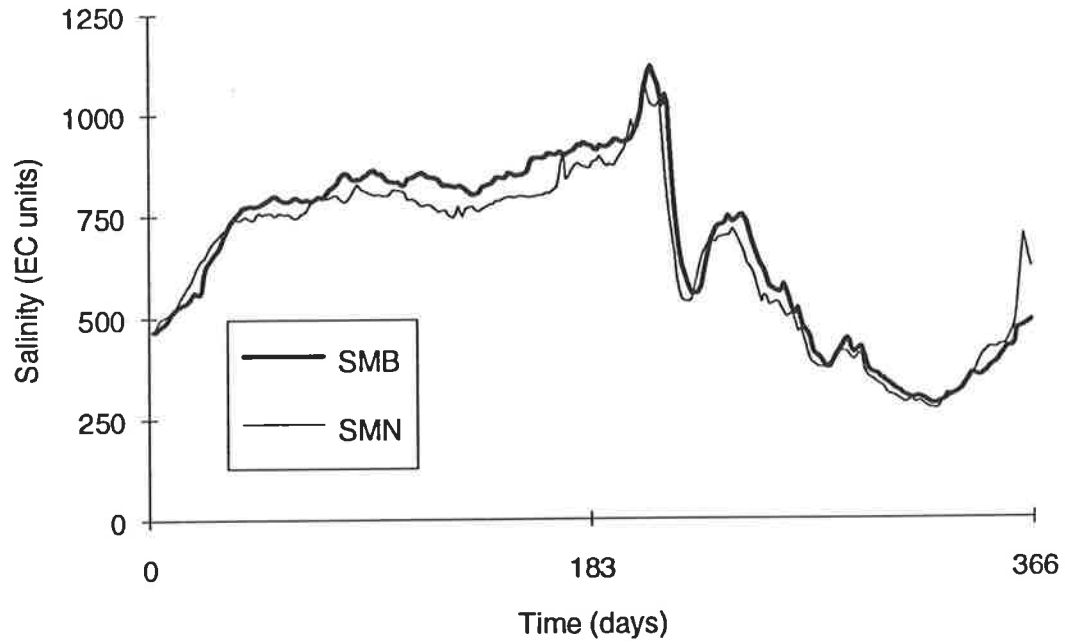


Figure 3.32: Salinity at Murray Bridge (SMB) and Salinity at Mannum (SMN) - 1991

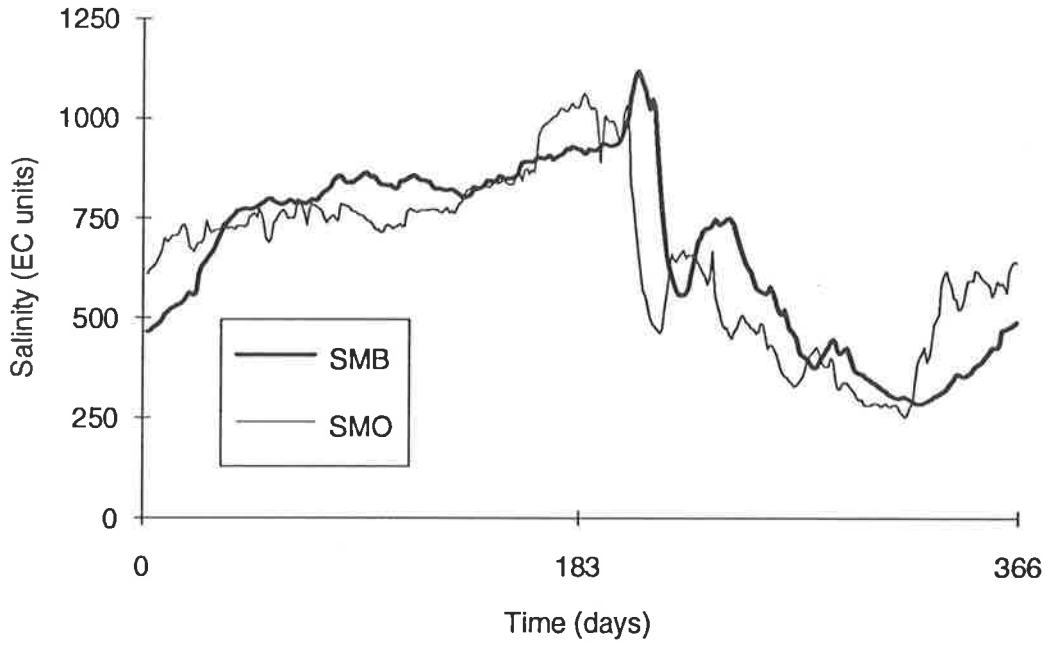


Figure 3.33: Salinity at Murray Bridge (SMB) and Salinity at Morgan (SMO) - 1991

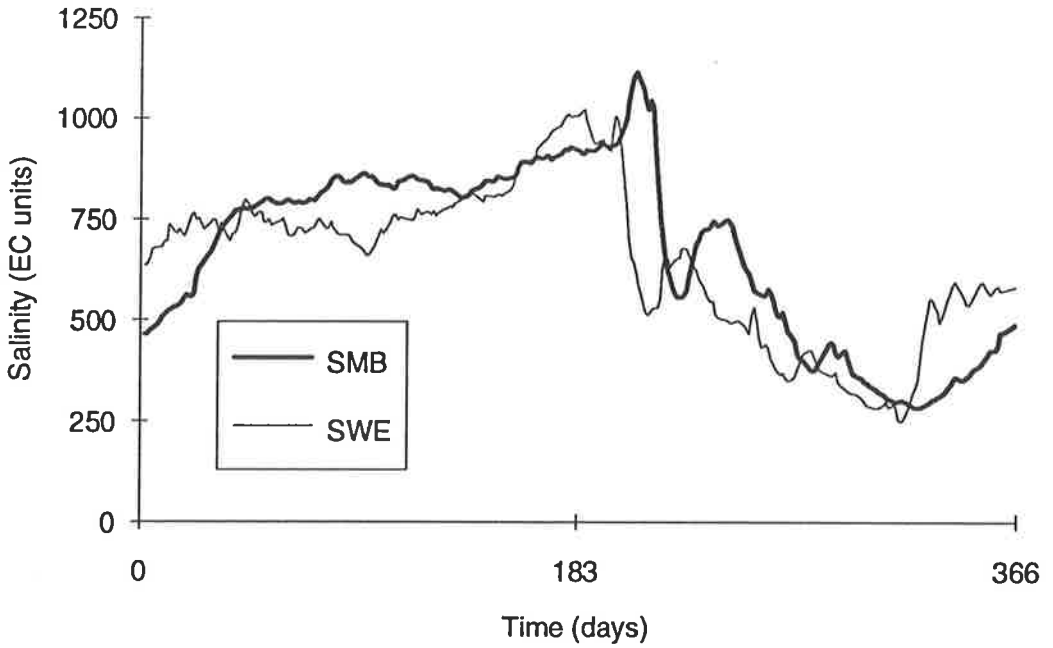


Figure 3.34: Salinity at Murray Bridge (SMB) and Salinity at Waikerie (SWE) - 1991

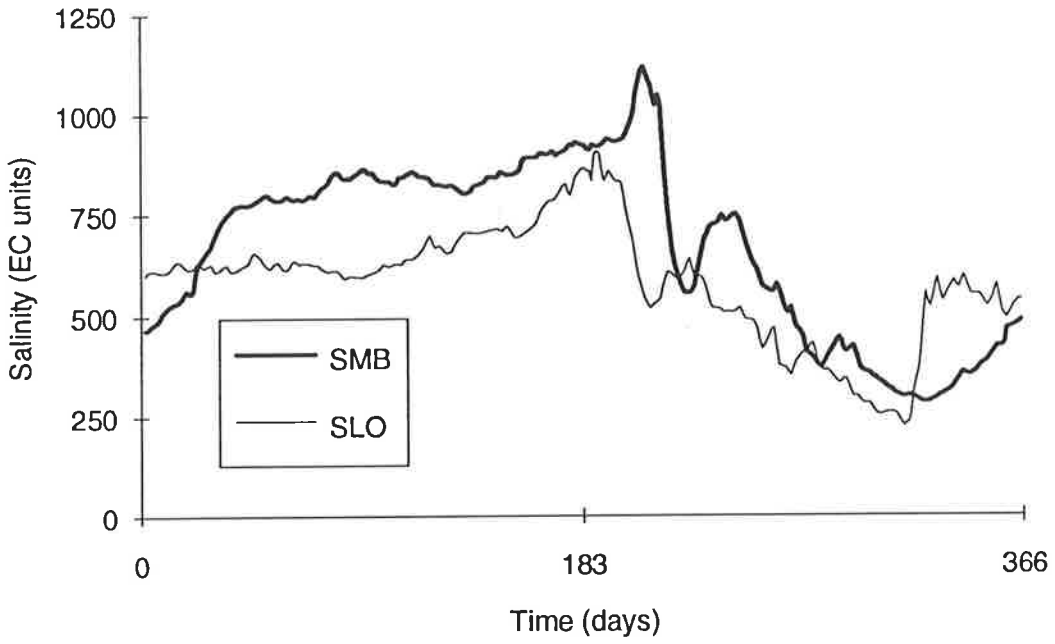


Figure 3.35: Salinity at Murray Bridge (SMB) and Salinity at Loxton (SLO) - 1991

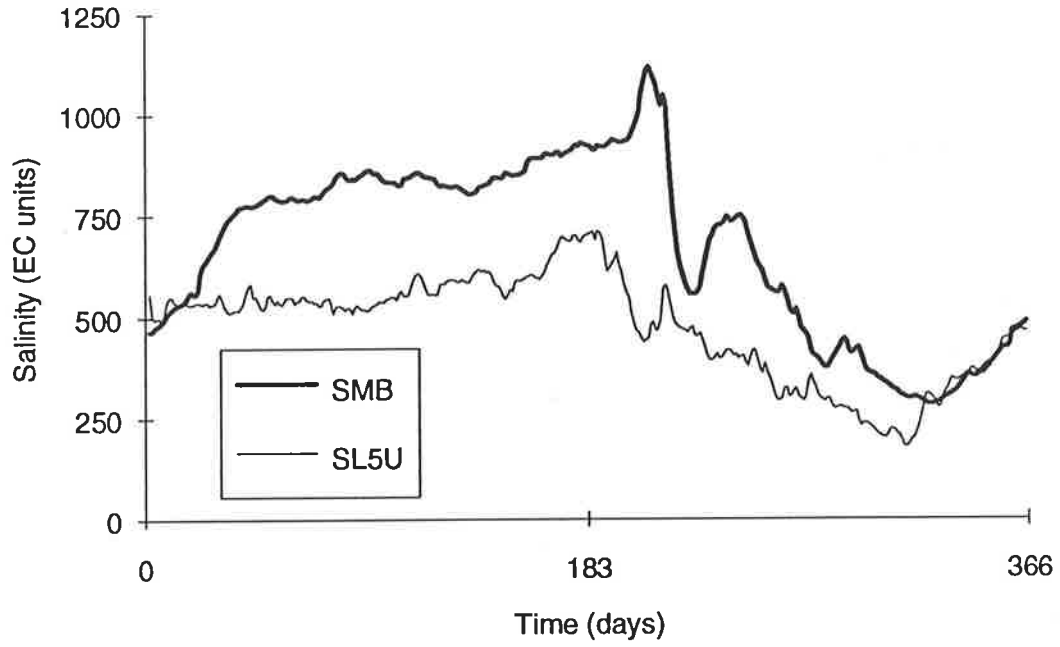


Figure 3.36: Salinity at Murray Bridge (SMB) and Salinity at Lock 5 Upper (SL5U) - 1991

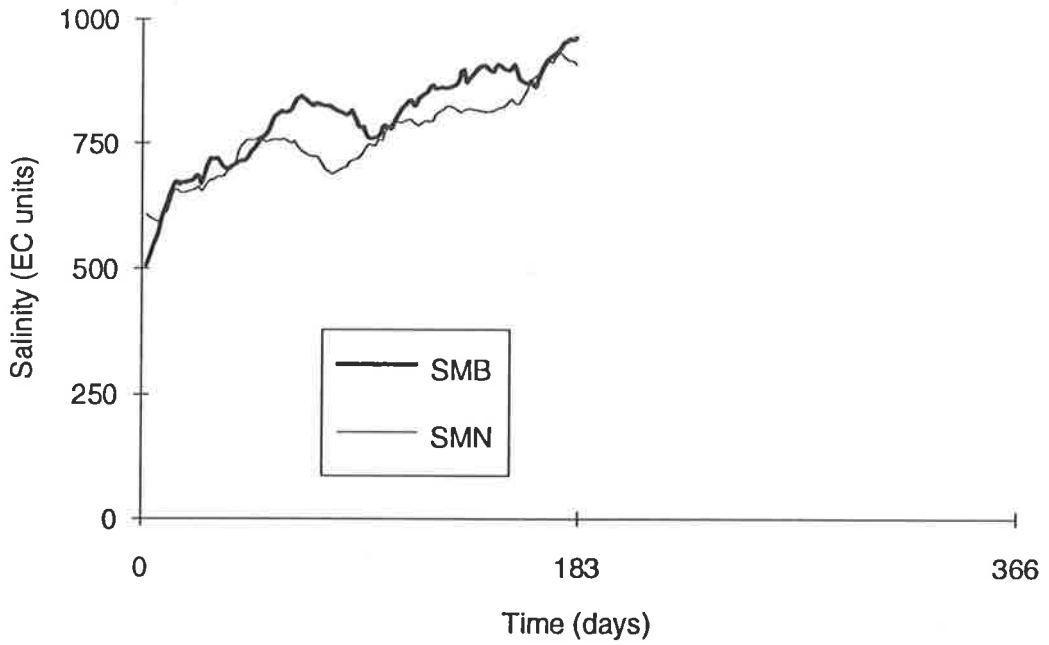


Figure 3.37: Salinity at Murray Bridge (SMB) and Salinity at Mannum (SMN) - 1992

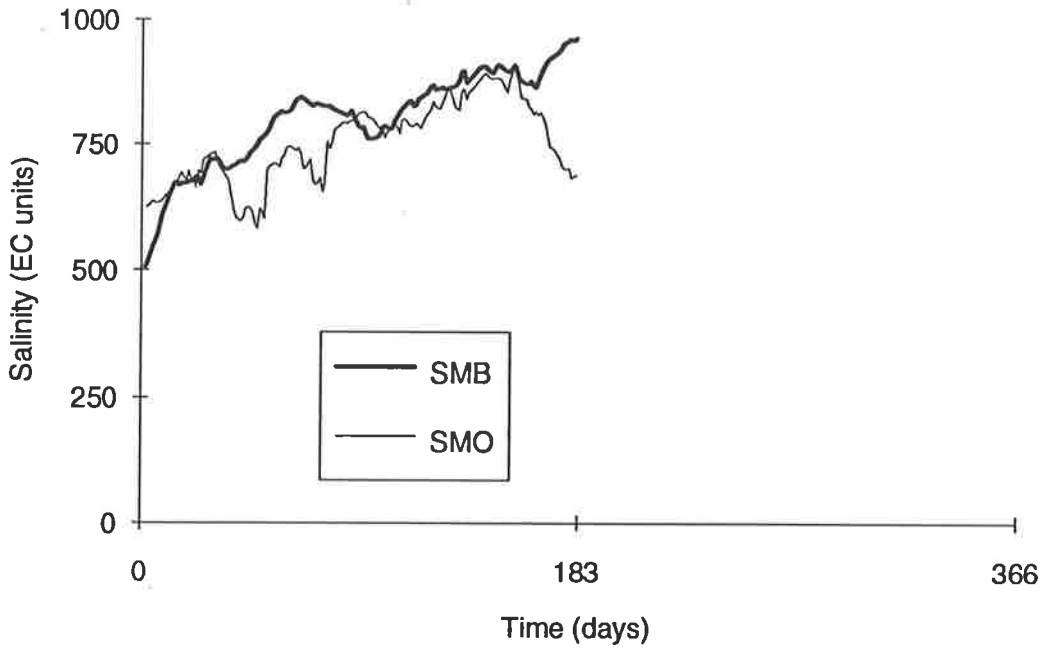


Figure 3.38: Salinity at Murray Bridge (SMB) and Salinity at Morgan (SMO) - 1992

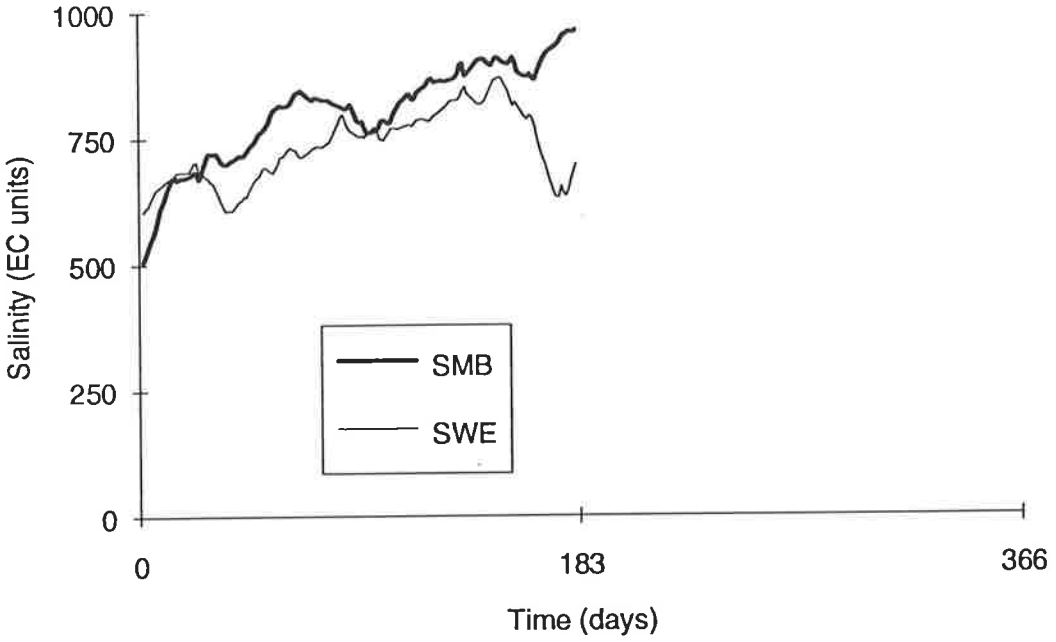


Figure 3.39: Salinity at Murray Bridge (SMB) and Salinity at Waikerie (SWE) - 1992

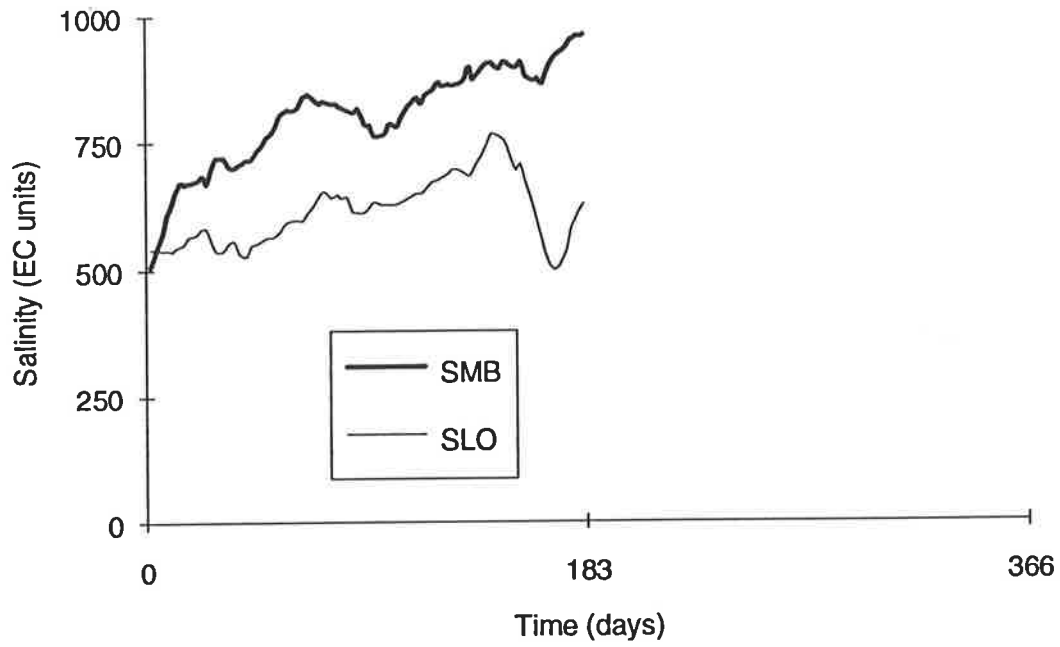


Figure 3.40: Salinity at Murray Bridge (SMB) and Salinity at Loxton (SLO) - 1992

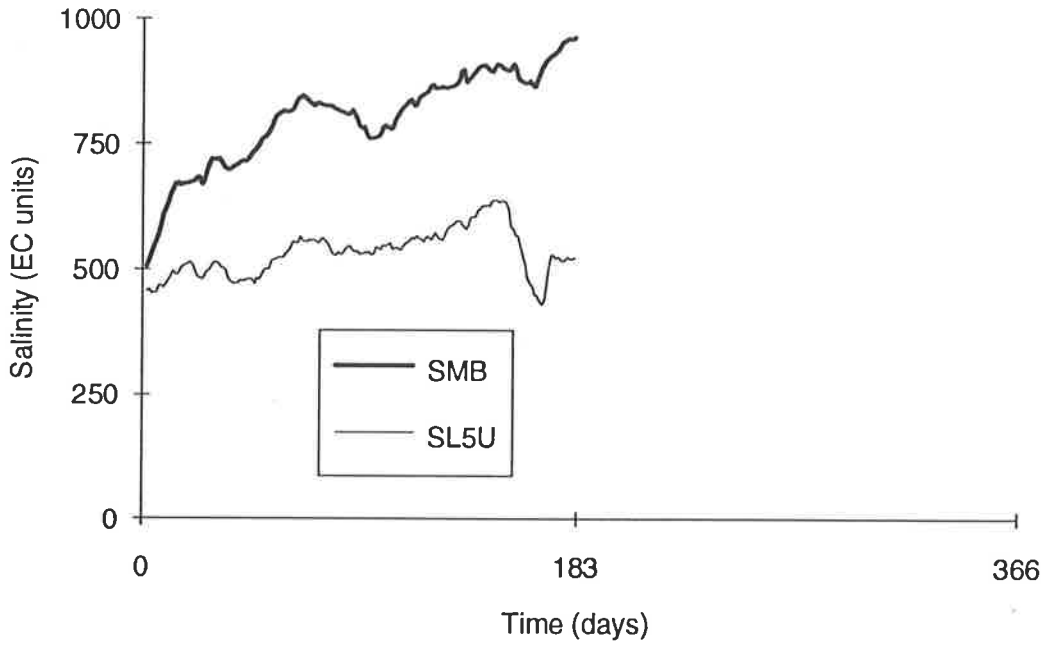


Figure 3.41: Salinity at Murray Bridge (SMB) and Salinity at Lock 5 Upper (SL5U) - 1992

Flow Data:

The means, standard deviations, maxima and minima of the flow time series are shown in Table 3.4. It can be seen that the summary statistics for flows at Lock 1 Lower and Overland Corner are very similar.

Table 3.4: Statistics of Flow Time Series (1987 to 1992)

Time Series	Average (ML/day)	Standard Deviation (ML/day)	Maximum (ML/day)	Minimum (ML/day)
FL1L	20378	25024	111000	1026
FOC	21891	24993	110618	1769

Plots of the flow time series and the salinity time series at Murray Bridge are shown in Figures 3.42 and 3.43. It can be seen that there is no long term trend and there appear to be no outliers. By examining the first four years of the time series (1987 to 1990), it appears as though there is an increasing trend in flow. However, the fact that the floods in successive years were bigger than the ones in the previous year is pure coincidence and confirmed by the lower flows in 1991. Both flow time series exhibit irregular seasonal variation, with floods occurring in the second half of the year. There also

seems to be quite a strong inverse relationship between flow and salinity at Murray Bridge.

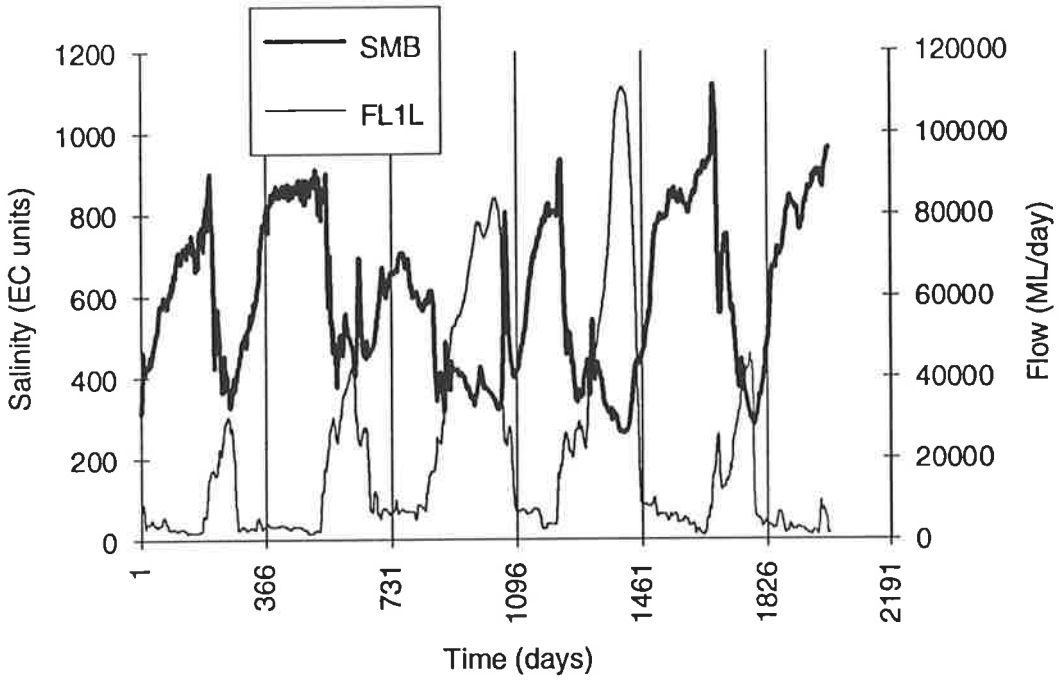


Figure 3.42: Salinity at Murray Bridge (SMB) and Flow at Lock 1 Lower (FL1L) - 1987 to 1992

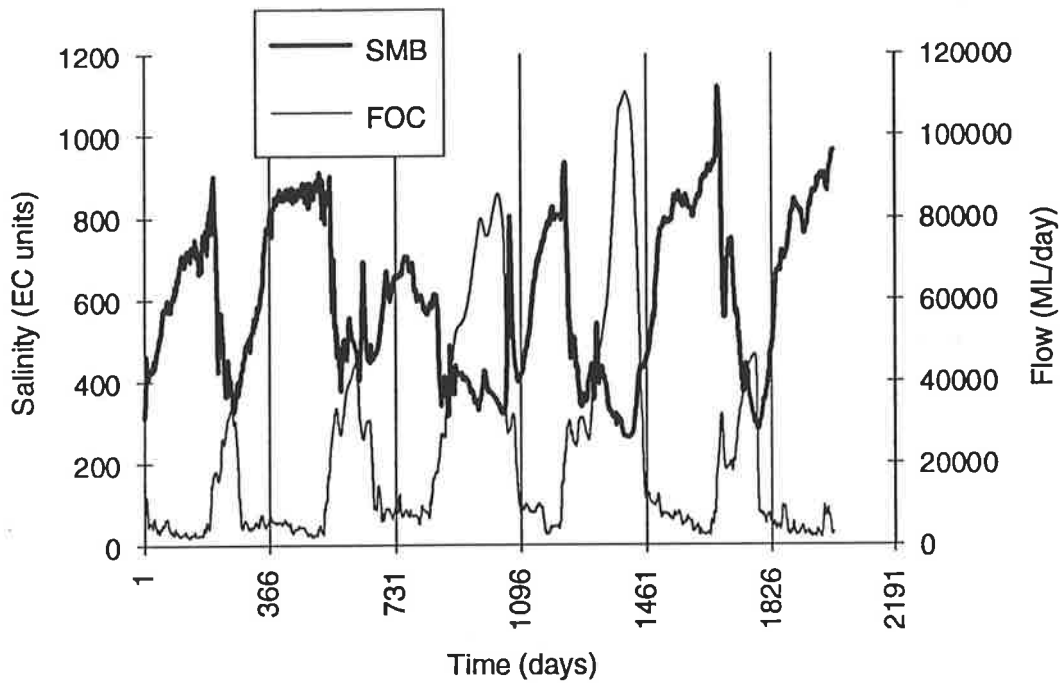


Figure 3.43: Salinity at Murray Bridge (SMB) and Flow at Overland Corner (FOC) - 1987 to 1992

River Level Data:

The means, standard deviations, maxima and minima of the river level time series are shown in Table 3.5. As expected, the mean and maximum values increase with increasing distance upstream.

Table 3.5: Statistics of River Level Time Series (1987 to 1992)

Time Series	Average (m)	Standard Deviation (m)	Maximum (m)	Minimum (m)
LMB	0.77	0.14	1.57	0.56
LMN	0.84	0.22	1.89	0.50
LL1L	1.61	1.16	5.28	0.50
LL1U	3.36	0.36	5.17	2.88
LMO	3.89	1.03	7.53	3.10
LWE	6.69	0.89	10.10	6.13
LOC	7.16	1.30	11.44	6.15
LLO	10.79	1.11	13.98	9.85

Plots of the river level time series and the salinity time series at Murray Bridge are shown in Figures 3.44 to 3.51. It can be seen that there is no long term trend and there appear to be no outliers. As is the case with the flow data, there appears to be an increasing trend over the first four years. However, this is not long term, for the reasons mentioned in the discussion of the flow data. As expected, there seems to be a strong positive correlation between flow and river levels, and hence an inverse relationship between river levels and salinity. This is particularly apparent for the levels at Lock 1 Lower, Morgan, Waikerie, Overland Corner and Loxton. The level at Lock 1 Upper is fairly constant, except for extreme flood cases. This is to be expected, as the aim of lock operations is to maintain a constant weir pool level. The levels at Murray Bridge and Mannum follow the general trend shown by the other river levels. However, the data appear to be very noisy.

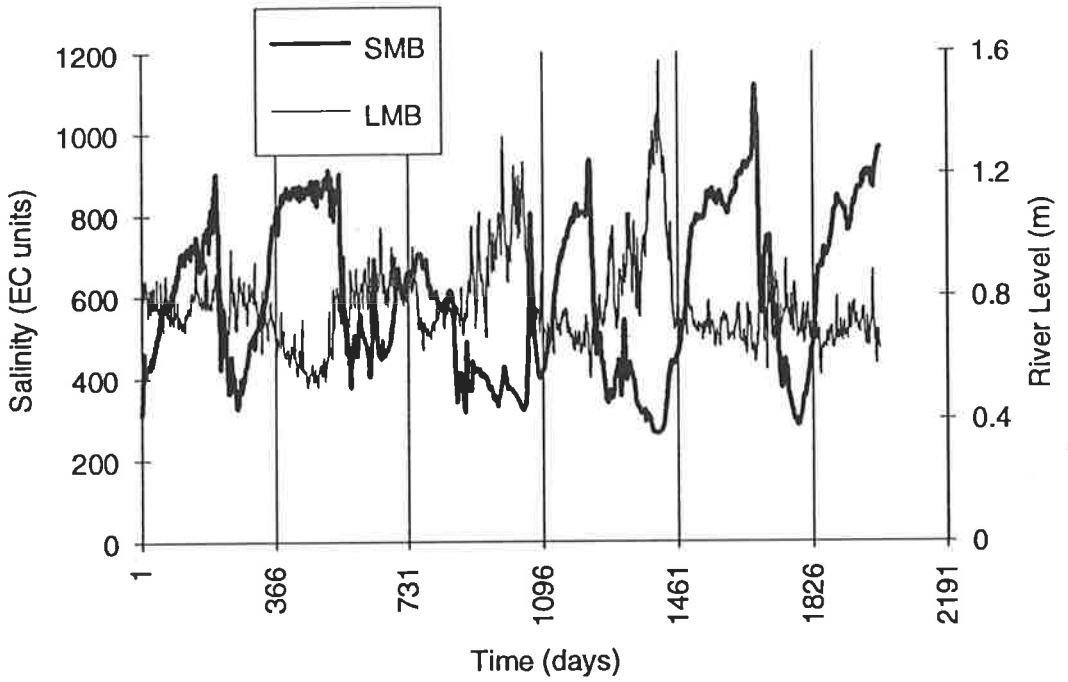


Figure 3.44: Salinity at Murray Bridge (SMB) and River Level at Murray Bridge (LMB) - 1987 to 1992

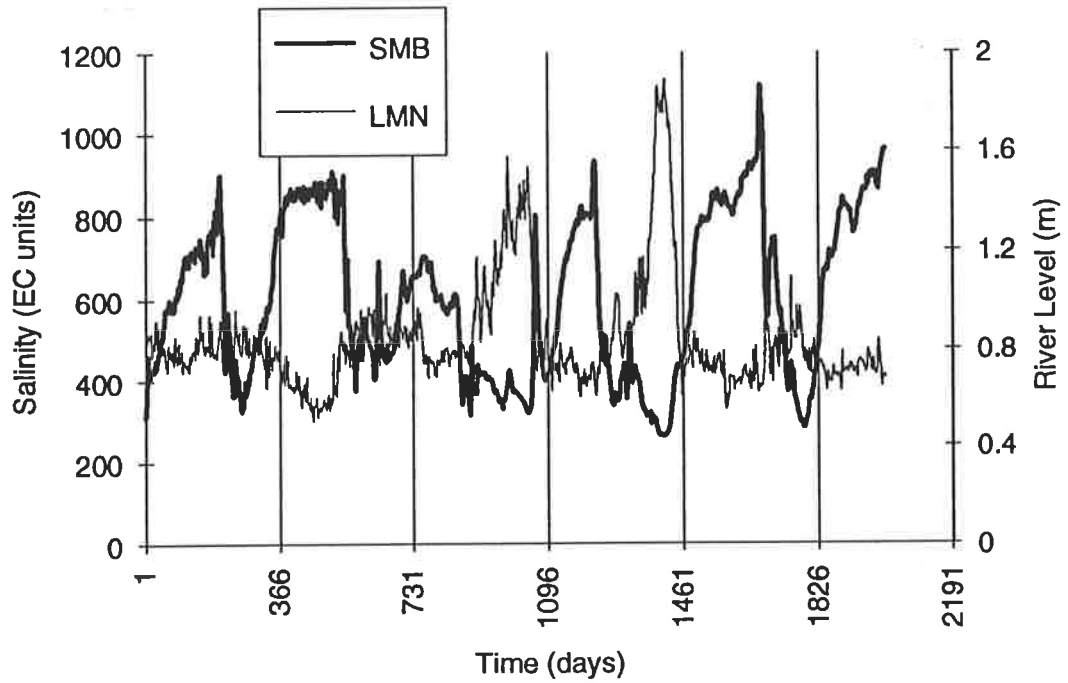


Figure 3.45: Salinity at Murray Bridge (SMB) and River Level at Mannum (LMN) - 1987 to 1992

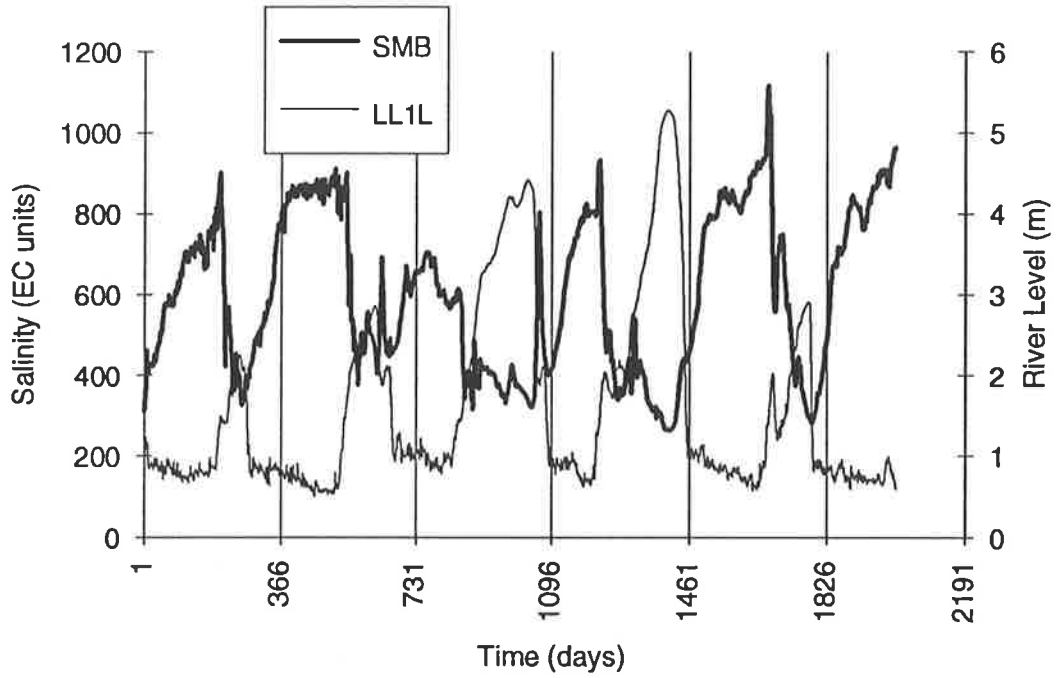


Figure 3.46: Salinity at Murray Bridge (SMB) and River Level at Lock 1 Lower (LL1L) - 1987 to 1992

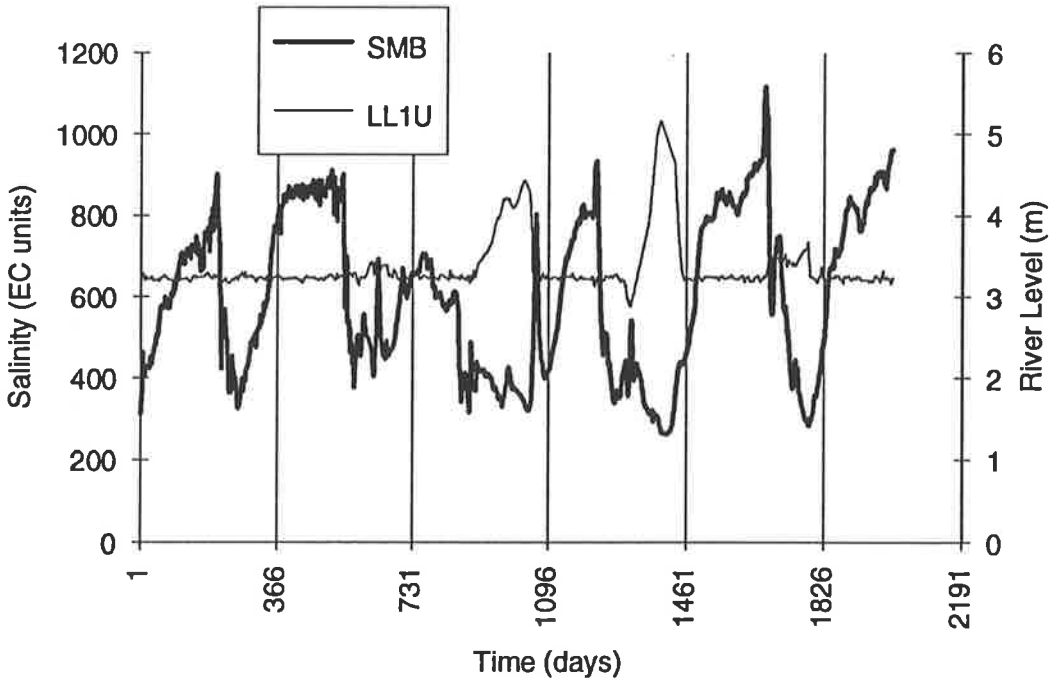


Figure 3.47: Salinity at Murray Bridge (SMB) and River Level at Lock 1 Upper (LL1U) - 1987 to 1992

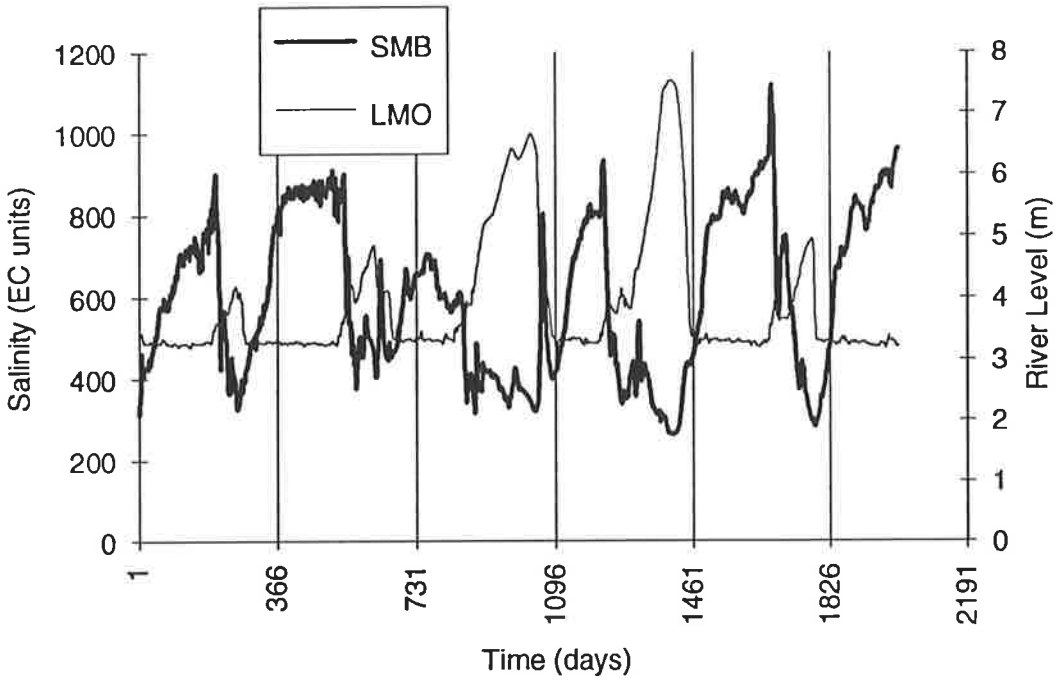


Figure 3.48: Salinity at Murray Bridge (SMB) and River Level at Morgan (LMO) - 1987 to 1992

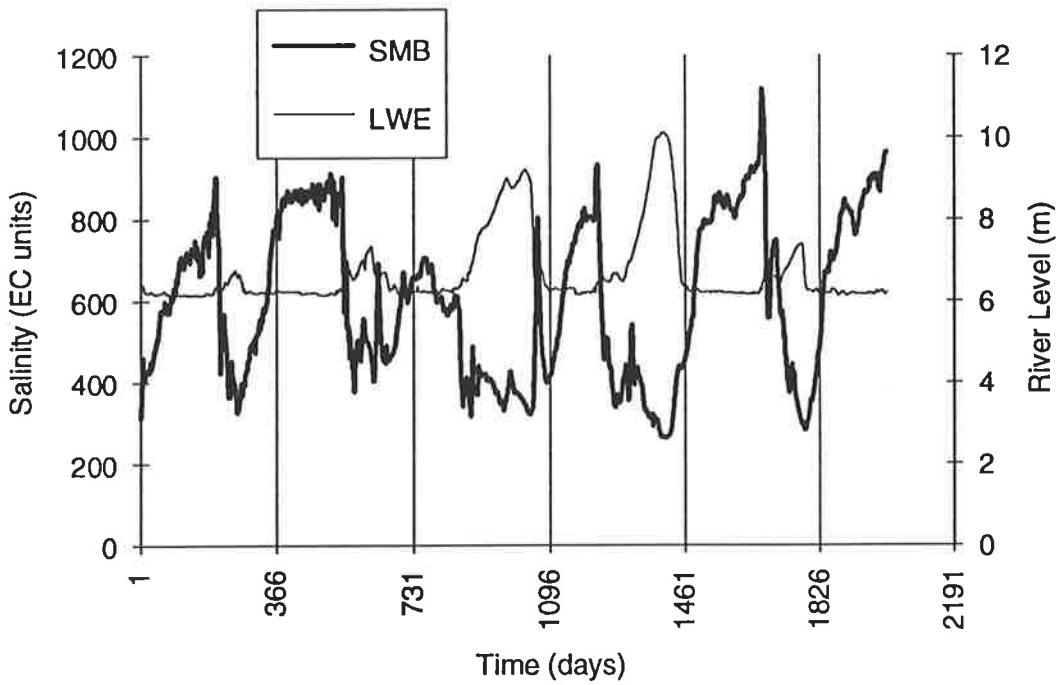


Figure 3.49: Salinity at Murray Bridge (SMB) and River Level at Waikerie (LWE) - 1987 to 1992

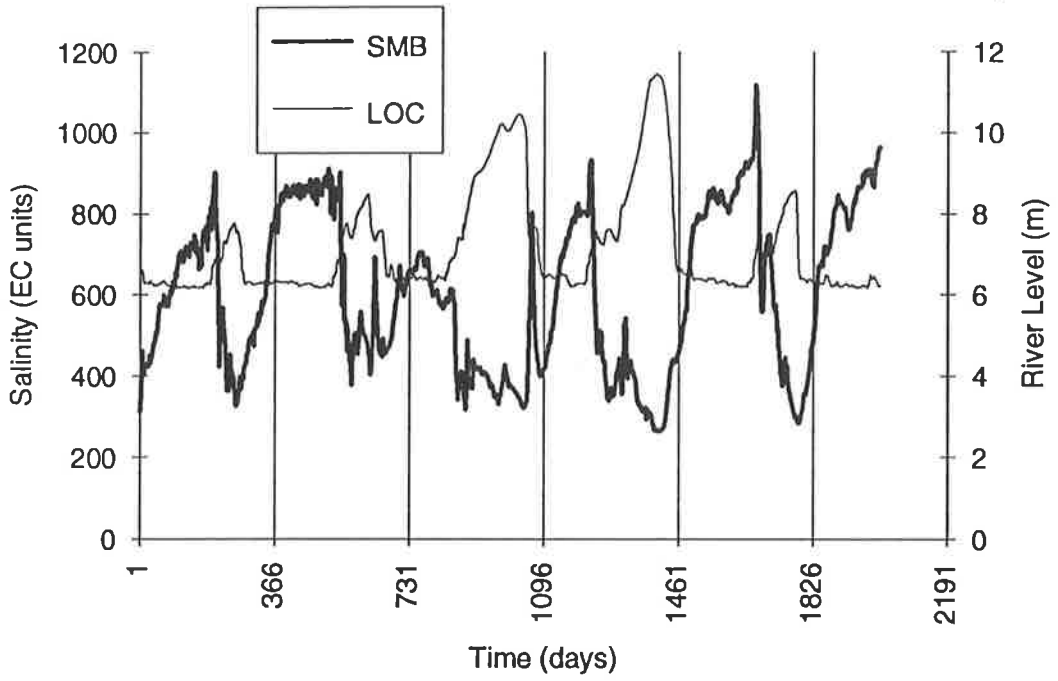


Figure 3.50: Salinity at Murray Bridge (SMB) and River Level at Overland Corner (LOC) - 1987 to 1992

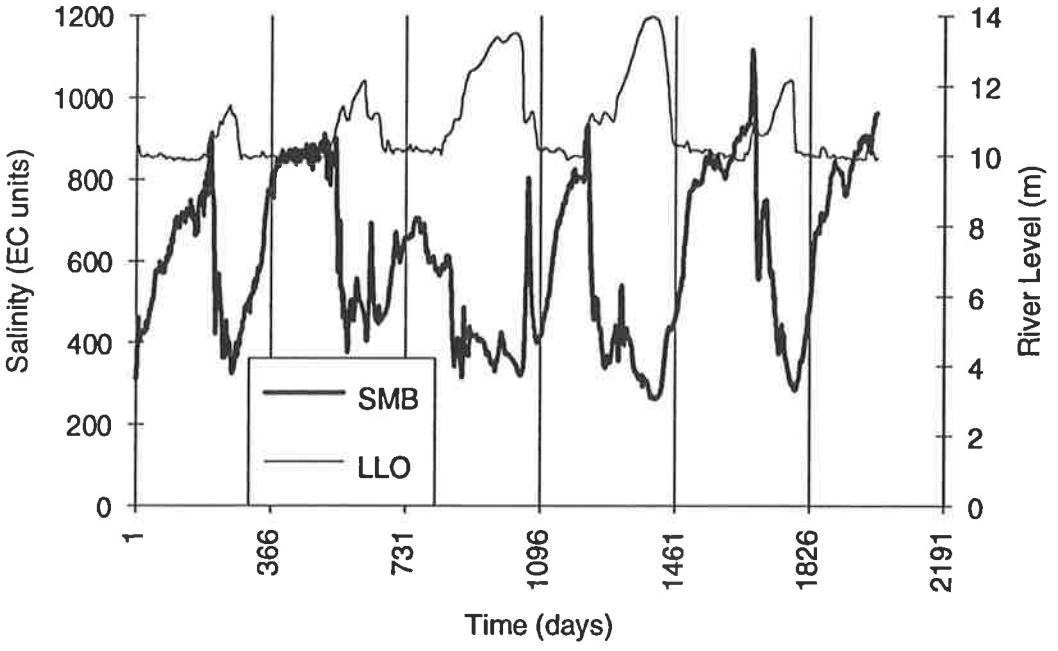


Figure 3.51: Salinity at Murray Bridge (SMB) and River Level at Loxton (LLO) - 1987 to 1992

3.3.4 Implementation

3.3.4.1 ANN Models

Development of the ANN models was carried out on a 80486 PC using the commercially available software packages NeuralWorks Explorer and NeuralWorks Professional II/Plus (NeuralWare, Inc., 1991). Some of the features of the software packages are described below:

1. The training and testing sets can be automatically scaled to a user defined range, producing **normalised** training and testing sets. The ranges used in conjunction with the linear and hyperbolic tangent transfer functions were -1.0 to +1.0 for the network inputs and -0.8 to +0.8 for the network outputs. The network output range was modified to +0.2 to +0.8 when the sigmoidal transfer function was used.
2. The number of nodes in the input layer, first hidden layer, second hidden layer, third hidden layer and output layer can be selected.
3. Different learning rates and momentum values can be selected for the different layers, and can also be changed as learning progresses.
4. There is a choice of three different learning rules, including the delta (DE) learning rule, the cumulative delta (CDE) learning rule and the normalised cumulative delta (NCDE) learning rule. The epoch size (ε) used in conjunction with the two cumulative learning rules can also be selected. It should be noted that when the NCDE learning rule is used, the error term in the weight update equation is equal to the individual errors produced by each training sample summed over one epoch, divided by the square root of the epoch size.
5. Three different transfer functions can be chosen, including linear, sigmoidal and hyperbolic tangent transfer functions.
6. There is a choice of three different error functions, including the quadratic, cubic and quartic error function.
7. There are facilities for using the delta-bar-delta (dbd) and extended delta-bar-delta (edbd) algorithms.
8. Learning progress can be monitored with the aid of a plot showing how the root mean squared error (RMSE) between the normalised actual and normalised predicted values of the training set varies during training.
9. During training, the network (connection weights) can be saved at predetermined intervals.
10. Knowledge can be extracted from a trained network by examining the connection weights and by carrying out a sensitivity analysis (using the software's "Explain" function), which determines how much each network output is changed in response to a given change in one of the network inputs. The effect of each network input is

examined in turn. The network inputs can be changed by a percentage of the input range of the network (see 1 above). A change of +5% was used in this research. The sensitivity of the inputs is given by Equation 3.1:

$$\text{Sensitivity} = \frac{\%\ \text{Change in Output}}{\%\ \text{Change in Input}} \times 100 \qquad (3.1)$$

11. The initial weight distribution is randomly generated within a range set by the user. The range chosen was -0.1 to +0.1 for all networks. Different random initial weight distributions (within the limits specified) can be obtained with the aid of the software's "Initialise" command.
12. During training, random values can be added to all connection weights (within a user specified range) to enable the network to escape local minima in the error surface. This process is referred to as "jogging" the connection weights.
13. There are no facilities for determining the optimimum network architecture using methods such as those described in Section 2.1.8.3, apart from very simple weight pruning techniques. Consequently, the optimum network geometry was obtained by trial and error in this research.

3.3.4.2 Time Series Models

The software required for the development of the ARIMA and VARIMA models was developed with the aid of a commercially available suite of Fortran subroutines (IMSL, 1991) in order to cater for daily data with seasonal variation. A list of the Fortran programs developed is given in Tables 3.6 to 3.8, indicating the name and function of each program, as well as the IMSL subroutines used.

Table 3.6: Computer Programs Developed for Time Series Modelling and their Function (Utility)

Program Name	IMSL Subroutines Used	Program Function
APCORR	ACF, PACF	• calculate ACF and associated bounds of significance • calculate PACF and associated bounds of significance • calculate the Box-Pierce statistic (Q^{BP})
DIFFER	DIFF	• difference a time series
CROSSCORR	CCF	• calculate CCF between two time series

Table 3.7: Computer Programs Developed for Time Series Modelling and their Function (Univariate)

Program Name	IMSL Subroutines Used	Program Function
PRELIM	NSPE	• calculate preliminary parameter estimates for an ARIMA model • determine whether the estimates are significant
PAREST	NSLSE	• calculate final least squares parameter estimates for an ARIMA model • calculate residuals
FCAST	NSBJF	• obtain differenced n-step Box-Jenkins forecasts • convert differenced forecasts into their original (non-differenced) form
UNIEST	FCN, UMACH, UMINF, MRRRR, LFTRG, LFGRG, LINRG, MXTYF	• calculate parameter estimates for ARIMA model in state space form • calculate log-likelihood and AIC
RESID	MRRRR	• calculate residuals of ARIMA model in state space form
PREDICT	MRRRR	• obtain differenced n-step forecasts for ARIMA model in state space form • convert differenced forecasts into their non-differenced form

Table 3.8: Computer Programs Developed for Time Series Modelling and their Function (Multivariate)

Program Name	IMSL Subroutine Used	Program Function
MULTIEST	FCN, UMACH, UMINF, MRRRR, LFTRG, LFGRG, LINRG, MXTYF	• calculate parameter estimates for VARIMA model in state space form • calculate log-likelihood and AIC
MPREDICT		• obtain differenced n-step forecasts for VARIMA model in state space form • convert differenced forecasts into their non-differenced form • calculate residuals

3.3.5 Performance Measures

The root mean squared error (RMSE) (Equation 2.21) was used to measure the generalisation ability of the models developed, as it places greater emphasis on larger forecasting errors. Consequently, models which have the ability to forecast rapid variations in salinity are favoured. The average absolute percentage error (AAPE) (Equation 2.22) and the average absolute error (AAE) (Equation 2.23) were also calculated. For the ANN models, training speed was also used as a performance measure.

The stopping criteria used for all networks was cross-validation (Section 2.1.8.1), which involves the following steps:

1. Start training by randomly presenting TI training samples to the network.
2. Stop training.
3. Assess network performance by presenting the test (validation) set to the network and calculating the RMSE between the actual and predicted values for the forecasting period of interest.
4. Continue training by randomly presenting a further TI training samples to the network (note: once all samples in the training set have been presented to the network, the training set is presented to the network repeatedly).
5. Repeat steps 2 to 4 until a plateau in the RMSE of the forecasts obtained using the test set has been reached.

The number of training samples presented to the network in one training / testing cycle (TI) will henceforth be referred to as the testing interval.

The test sets used usually consist of one year of daily data. Consequently, when the test set is presented to a model, 365 forecasts are obtained. The performance of each model was measured by averaging the RMSE (and the AAPE and AAE) over the 365 forecasts (i.e. $P^{TE} = 365$ in Equations 2.21 to 2.23). The RMSEs of the forecasts obtained using the test set are shown in Appendix E for the various models developed.

3.3.6 Nomenclature

The following nomenclature will be used in this chapter to identify the different models developed:

Model Name:	X	X	X	-	X	-	X	-	X	X	-	X	X	-	X	X
Cell Number:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Cell Number(s):	Meaning:
1-3	Identify which time series is being forecast. The abbreviations used are in accordance with Table 3.2 (eg. SMB = Salinity at Murray Bridge LOC = Level at Overland Corner).
5	Identify whether the model is univariate or multivariate. U = Univariate M = Multivariate
7	Identify model Type. A = ANN model T = Time Series Model
9-10	ANN model: model number (01 to 99). Identify internal network parameters (e.g. learning rate, momentum, learning rule, epoch size, error function, transfer function) and network geometry (i.e. the number of hidden layers and the number of nodes per hidden layer). Parameters and geometries corresponding to each model number are given in Appendix D. Time Series model: model number (01 to 99). Identify type and order of model parameters.

12-13 ANN model: training / test set (01 to 99).
Identify network inputs and outputs. The inputs and outputs corresponding to each training / testing set are given in Appendix C.

Time Series model: transformed data set identification code.
Identify the degree of differencing of the data set used for model development.
Cell 4: Degree of seasonal differencing (D = 0, 1 or 2).
Cell 5: Degree of non-seasonal differencing (d = 0, 1 or 2).

15-16 Identify the year of data used for forecasting.
(eg. 91 = 1991
87 = 1987)

3.4 Development of Univariate Neural Network Model

3.4.1 Introduction

In this section, the development of the univariate ANN (UANN) model for the prediction of salinity in the River Murray at Murray Bridge is described. The objectives are:

1. To develop a UANN model which provides a yardstick against which the performance of the more complex multivariate ANN (MANN) model can be compared.
2. To address the following issues in relation to ANN models:
   - Is it better to train ANNs using stationary data or raw data?
   - When forecasting for more than one time step, is it better to train the model for a single time step forecast and then apply it recursively, or train it for a multistep forecast directly?
   - What is the effect of adding less significant inputs on network performance?
   - What is the effect of including hidden layers?

The steps shown in Figure 2.18 were followed to develop the various UANN models. Initially, the performance of the various models was evaluated for short (1 day) as well as longer term (14 day) forecasts. A real time forecasting situation was also simulated, in which 1, 5 and 14 day forecasts were obtained. This enables the performance of the

various types of models (i.e. univariate ANN, multivariate ANN, univariate time series and multivariate time series) to be compared for various forecasting periods.

For all models developed, data from 1987 to 1990 were used for training and data from 1991 were used for testing in order to simulate real time forecasting conditions. It was decided to use the 1991 data for testing, rather than the data from the last half of 1991 and the first half of 1992, as most of the major variations in salinity occur in the middle of the year.

3.4.2 Checking of the Time Series for Stationarity

One of the objectives when developing the UANN model is to determine whether it is better to use stationary data or raw data. As explained in Section 3.3.3.1, the salinity time series at Murray Bridge exhibits a distinct seasonal pattern. Consequently, one would expect the time series to be non-stationary. In this section, the statistical tests for stationarity described in Section 2.2.6 were applied to the time series to determine whether the assumption of non-stationarity is valid.

The ACF and the PACF of the salinity time series at Murray Bridge were obtained with the aid of the computer program APCORR. The maximum lag used was 1095, which is equivalent to 3L (i.e. 3 times the seasonal component of 365 days). Bowermann and O'Connell (1979) suggest using a maximum lag of 4L. However, due the limited data available, the maximum lag that could be used was 3L. The ACF and PACF are shown in Figures 3.52 and 3.53 respectively. As can be seen in Figure 3.52, the ACF dies down extremely slowly, indicating that the original time series is non-stationary. The seasonality of approximately 365 days is confirmed by a strong negative correlation at approximately 183 days (0.5L) and a strong positive correlation at approximately 365 days (L). The PACF dies down very quickly, as shown in Figure 3.53.

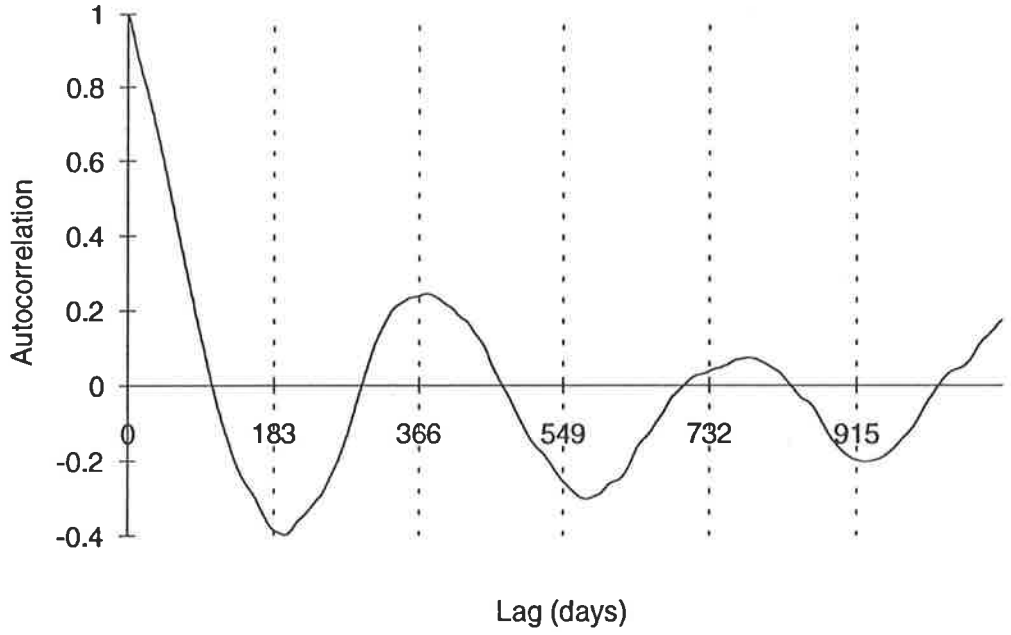


Figure 3.52: ACF for Salinity at Murray Bridge (1987-1991)

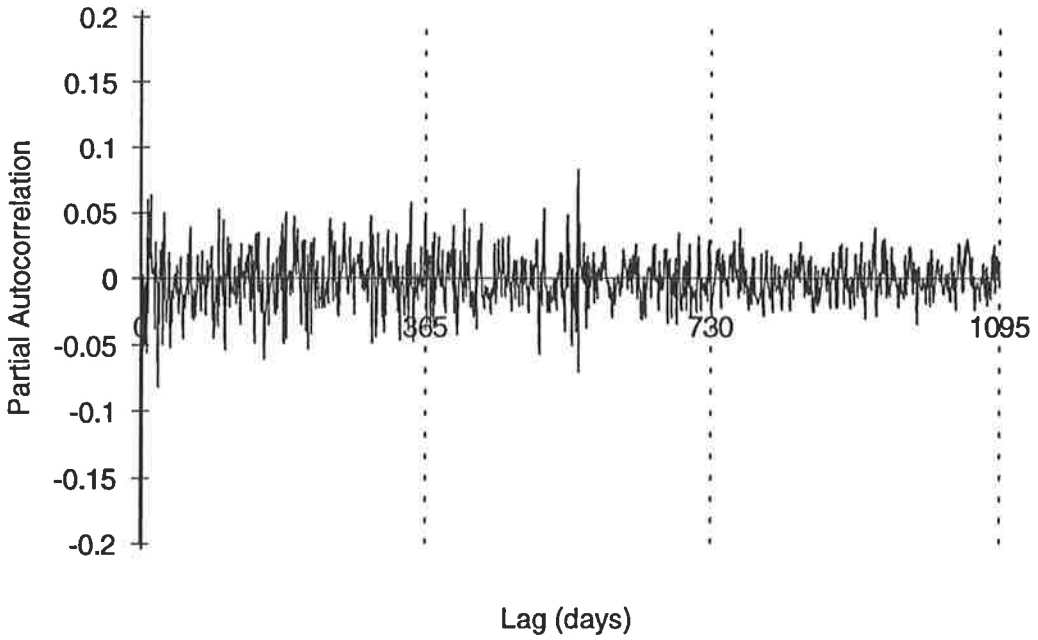


Figure 3.53: PACF for Salinity at Murray Bridge (1987-1991)

3.4.3 Transformation of the Data by Differencing

As the original time series is non-stationary, it was transformed into a stationary time series using differencing (Box and Jenkins, 1976). Differencing is the simplest way to remove trends and seasonalities (Vandaele, 1983). One problem associated with differencing is that the transformed time series may possess infinite variance (Irvine and Eberhardt, 1992). Using more sophisticated methods, such as harmonic analysis with Fourier coefficients, may overcome this problem (Tao and Delleur, 1976). However, as well as being more time consuming, more complex methods can still induce analytical distortions (Salas et al., 1980). Consequently, it was decided to use differencing in this research.

Differencing was carried out in accordance with Equation 2.56, as the time series exhibits seasonal fluctuations. The seasonality (L) was chosen to be 365 days and the various degrees of seasonal and non-seasonal differencing used are summarised in Table 3.9.

Table 3.9: Degrees of Seasonal and Non-Seasonal Differencing Used

Transformed Data Set Identification Code (dD)	Degree of Non-Seasonal Differencing (d)	Degree of Seasonal Differencing (D)
10	1	0
20	2	0
01	0	1
02	0	2
12	1	2
21	2	1
11	1	1
22	2	2

Computer program DIFFER was used to perform the differencing operation. As discussed in Section 3.4.1, it was decided to use the salinity data from 1987 to 1990 for parameter estimation purposes. As a result, only data from 1987 to 1990 were differenced and examined for stationarity at this stage. The ACF and the PACF for each differenced data set were obtained using computer program APCORR. The maximum lag used was 1095. In the cases where the degree of seasonal differencing used was 2 (i.e. D=2), the data set was reduced to fewer than 1095 values due to the large number of values lost as part of the differencing process. In these cases, the largest available lag was used. The ACF and the PACF of the differenced data sets were examined for non-

zero values in accordance with Equations 2.40 and 2.43 respectively. The bounds of significance were automatically obtained by computer program APCORR. The lags at which the values of the ACF and the PACF were significantly different from zero are shown in Tables 3.10 to 3.12. The numbers in brackets indicate the total number of significant values.

It should be noted that the only values of the ACF and PACF that were substantially greater than the bounds of significance were those at very low lags (i.e. 1 to 10), and those in the vicinity of one seasonal lag (i.e. 364 to 367). The other values that were significantly different from zero were in the vicinity of the bounds of significance.

Table 3.10: Lags at which Values of the ACF and PACF are Significantly Different from Zero for the Differenced Data Sets Without Seasonal Differencing (D=0)

Transformed Data Set Identification Code	Lags at which significant values occur (days)	
	ACF	PACF
10	1, 2, 3, 4, 12, 21, 22, 23, 177, 179, 180, 201, 311, 312, 427, 428. (16)	1, 2, 12, 21, 26, 27, 29, 37, 62, 98, 104, 106, 140, 141, 142, 143, 156, 179, 196, 199, 251, 259, 274, 300, 310, 331, 345, 348, 365, 405, 427, 511. (32)
20	1, 2, 3, 12, 17, 20, 21, 26, 28, 36, 96, 97, 144, 180, 364, 365, 369. (17)	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 14, 15, 16, 19, 20, 22, 26, 31, 36, 38, 52, 55, 61, 64, 97, 103, 105, 140, 141, 142, 155, 178, 250, 258, 298, 299, 309, 330, 344, 347, 364, 426, 510. (45)

Table 3.11: Lags at which Values of the ACF and PACF are Significantly Different from Zero for the Differenced Data Sets With a Degree of Seasonal Differencing Equal to 1 (D=1)

Transformed Data Set Identification Code	Lags at which significant values occur (days)	
	ACF	PACF
01	Dies down extremely slowly.	1, 2, 63, 153, 183, 186, 189, 197, 211, 350, 365, 366, 561. (13)
11	1, 2, 3, 10, 12, 13, 26, 62, 63, 64, 144, 180, 364, 365, 366. (14)	1, 2, 8, 11, 12, 21, 25, 26, 29, 37, 40, 57, 62, 73, 82, 89, 93, 104, 127, 132, 143, 185, 205, 251, 299, 300, 304, 310, 320, 347, 348, 364, 365, 366. (34)
21	1, 2, 3, 8, 10, 12, 21, 26, 28, 36, 40, 56, 93, 106, 131, 142, 144, 180, 182, 309, 329, 363, 365. (23)	1, 2, 3, 4, 5, 6, 8, 9, 12, 14, 16, 19, 20, 22, 26, 27, 29, 36, 48, 52, 55, 61, 64, 65, 81, 92, 98, 104, 105, 126, 142, 184, 250, 298, 299, 303, 309, 319, 346, 347, 363, 364, 365. (43)

Table 3.12: Lags at which Values of the ACF and PACF are Significantly Different from Zero for the Differenced Data Sets With a Degree of Seasonal Differencing Equal to 2 (D=2)

Transformed Data Set Identification Code	Lags at which significant values occur (days)	
	ACF	PACF
02	Dies down extremely slowly.	1, 2, 26, 27, 38, 63, 186, 189, 197, 366, 367. (11)
12	1, 2, 12, 26, 62, 63, 64, 144, 364, 365, 366. (11)	1, 2, 11, 12, 21, 25, 29, 37, 40, 57, 62, 73, 82, 89, 127, 132, 133, 150, 178, 304, 364, 365, 366, 367. (24)
22	1, 2, 3, 10, 12, 26, 28, 36, 38, 40, 56, 131, 142, 144, 180, 182, 365. (17)	1, 2, 3, 4, 5, 8, 9, 12, 14, 16, 19, 20, 36, 42, 48, 56, 61, 72, 81, 88, 98, 125, 132, 149, 299, 303, 363, 364, 365, 366. (30)

Bowermann and O'Connell (1979) suggest that seasonal data may be deemed stationary if the ACF and/or the PACF cut off at a lag less than or equal to 2L+2. However, if either the ACF or PACF die down extremely slowly, the time series must be deemed non-stationary. Using these criteria, all differenced data sets, with the exception of data sets 01 and 02, may be deemed stationary. The PACFs of all differenced time series cut off at a lag that is less than 732 (i.e. 2L+2). In addition, the PACFs for data sets 02, 12, 21, 11 and 22 cut off at a lag that is less than or equal to 367 (i.e. L+2). The ACFs of data sets 10, 20, 12, 21, 11 and 22 also cut off at a lag that is less than or equal to 732. In addition, the ACFs for data sets 12, 21, 11 and 22 cut off at a lag that is less than or equal to 367. As the PACFs, as well as the ACFs, cut off at a lag that is less than 2L+2 for data sets 10, 20, 12, 21, 11 and 22, they may be considered stationary. The ACFs for data sets 01 and 02, on the other hand, die down extremely slowly, and despite the fact that the PACFs of these data sets cut off at a lag that is less than 732, they must be considered non-stationary. This implies that seasonal differencing alone is not enough to remove non-stationarities.

Theoretically, any of the stationary data sets could have been used in the development of the UANN model. However, it was decided to only use one of the stationary data sets for model development. The following criteria were considered for choosing the most appropriate stationary data set:

- The lowest appropriate degree of differencing should be used, as differencing might induce noise.
- As the raw data has distinct seasonal variation, some degree of seasonal differencing appears to be appropriate.

Taking these criteria into consideration, data set 11 was chosen. A Plot of the transformed data set is shown in Figure 3.54. Plots of the ACF and the PACF of this set, including the bounds of significance, are shown in Figures 3.55 and 3.56. It can be seen that the variance of the data set does not appear to be constant, and that the values of the ACF and PACF in the vicinity of one seasonal lag are substantially greater than the bounds of significance. However, as the ACF and the PACF cut off at lags less than or equal to 732, data set 11 may be deemed stationary (Bowermann and O'Connell, 1979).

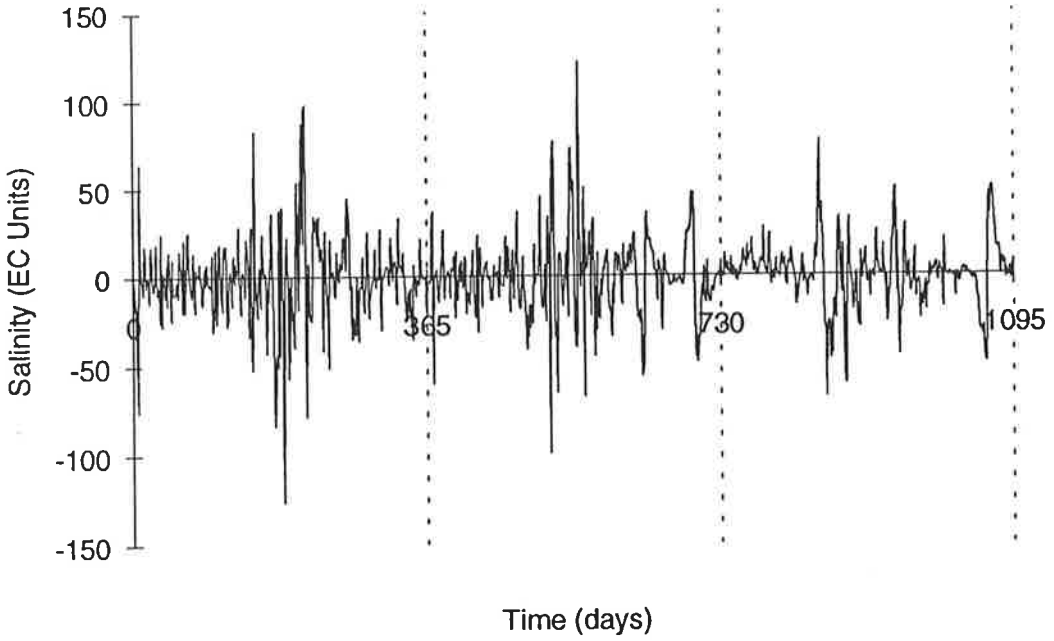


Figure 3.54: Data Set 11

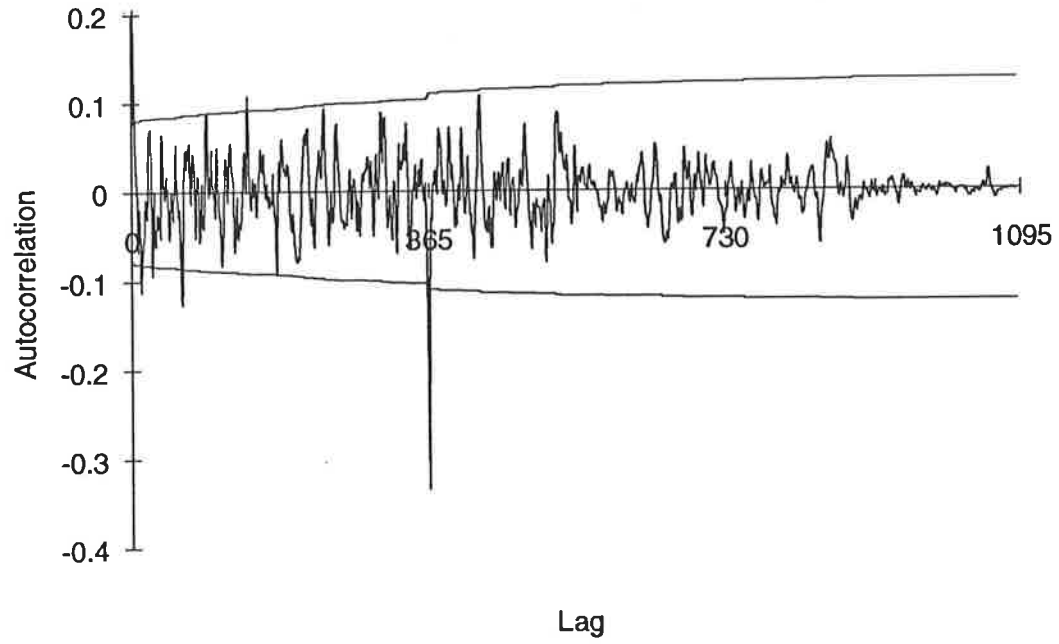


Figure 3.55: ACF for Data Set 11

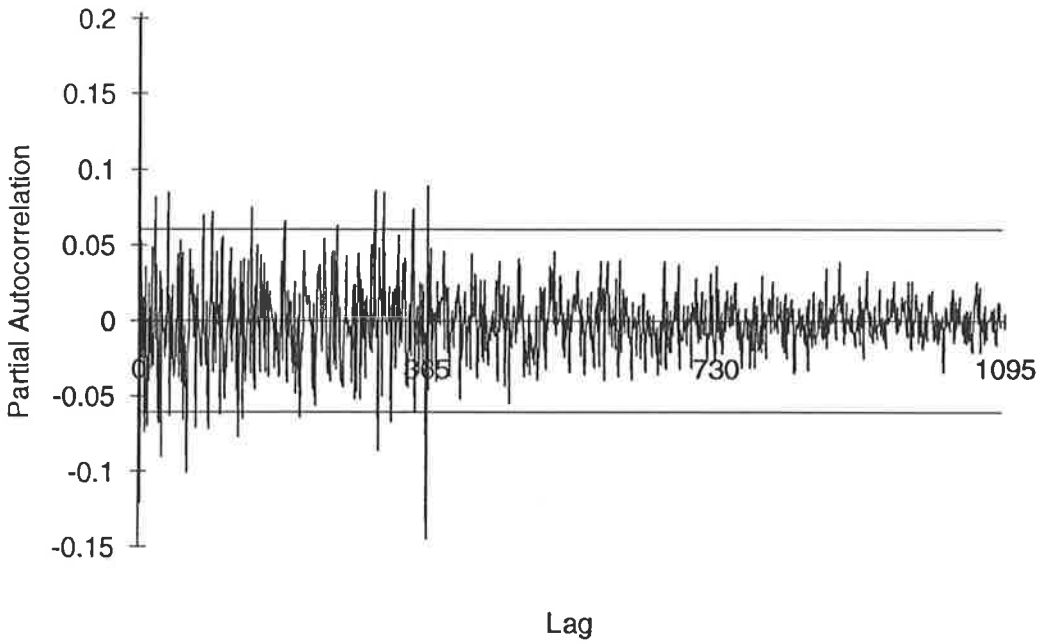


Figure 3.56: PACF for Data Set 11

3.4.4 Development of Models Using the Differenced Data

In this section, all networks were trained and tested using transformed data set 11. The differenced forecasts obtained (u_t) were converted to their original (non-differenced) form (z_t) using the following equation:

$$z_t = u_t + z_{t-1} + z_{t-365} - z_{t-366} \tag{3.2}$$

Forecasts were obtained for forecasting periods of one and 14 days. It should be noted that a recursive procedure had to be used to obtain the multi step forecasts (z_{t+g}), as the original (non-differenced) form of the (g-1)-step forecast (z_{t+g-1}) is required to transform the differenced form of the n-step forecast (u_{t+g}) into its non-differenced form (i.e. z_{t+g}) (see Equation 3.2).

The effect of network geometry and the number of network inputs on network performance was also investigated. The learning rate (η) used was 0.02, the momentum value (μ) used was 0.6 and the epoch size (ε) used was 5 for all models unless stated otherwise. Details of the network geometry and the internal parameters used for the various models are given in Appendix D. Details of the training / testing sets used for the various models are given in Appendix C. The number of training samples presented to each network was 100,000 and the testing interval used was 10,000 unless stated

otherwise. The performance of the models at the various stages of learning is given in Appendix E. As indicated in Section 3.3.5, the values shown in Appendix E are the RMSEs between the actual and predicted values of the **testing** sets.

3.4.4.1 Determination of Initial Model Inputs

As the stationary data were used for model development, the ACF and PACF could be used as a guide for determining the initial model inputs. The largest values in the ACF and PACF occur at low lags (1 to 3) and in the vicinity of one seasonal lag (364 to 367). Consequently, the lags of the initial inputs were chosen to be 1, 2, 3, 364, 365, 366 and 367 (training / testing set 01).

It was decided to check which of the initial inputs were dominant by training a simple network without hidden layers and a linear transfer function, as in such a network, the strength of the connection weights gives a direct indication of the significance of each of the inputs. It should be noted that a network with this configuration is operationally similar to a univariate AR model, as the network output is close to a linear combination of the network inputs.

Different learning rates were used to determine whether the learning rate affects the connection weights obtained. The learning rates used include 0.001 (model SMB_U_A_01_01_91), 0.005 (model SMB_U_A_02_01_91) and 0.02 (model SMB_U_A_03_01_91). The weights corresponding to each input were monitored as learning progressed. Training was continued until the connection weights stabilised. The connection weights at the different stages of learning are shown in Tables 3.13 to 3.15. It can be seen that the weights stabilised at approximately the same values, regardless of the learning rate used. The variation of the connection weights with increased learning was slightly greater when larger learning rates were used. However, the weights stabilised much more quickly when larger learning rates were used (e.g. at a learn count of approximately 20,000 for a learning rate of 0.02 as compared with a learn count of approximately 360,000 for a learning rate of 0.001). As indicated in Tables 3.13 to 3.15 and Figure 3.57, the inputs at lags 1, 365 and 366 were clearly dominant. Consequently, it was decided to use a learning rate of 0.02 in conjunction with inputs at lags 1, 365 and 366 (training / testing set 02) for the networks trained with the differenced data.

Table 3.13: Connection Weights at Different Stages of Learning - Model SMB_U_A_01_01_91 ($\eta = 0.001$)

Learn Count	Lag of inputs						
	1	2	3	364	365	366	367
30,000	0.316	0.087	0.005	-0.079	-0.289	0.035	0.046
60,000	0.406	0.074	-0.023	-0.067	-0.238	0.099	0.034
160,000	0.491	0.024	-0.041	-0.055	-0.258	0.163	-0.007
260,000	0.508	0.004	-0.038	-0.055	-0.261	0.172	-0.002
360,000	0.514	-0.002	-0.036	-0.053	-0.260	0.175	-0.018
410,000	0.514	-0.002	-0.035	-0.053	-0.260	0.175	-0.019
460,000	0.514	-0.003	-0.036	-0.053	-0.260	0.176	-0.020
510,000	0.515	-0.004	-0.036	-0.055	-0.262	0.176	-0.018
560,000	0.514	-0.004	-0.035	-0.054	-0.259	0.177	-0.020

Table 3.14: Connection Weights at Different Stages of Learning - Model SMB_U_A_02_01_91 ($\eta = 0.005$)

Learn Count	Lag of inputs						
	1	2	3	364	365	366	367
10,000	0.364	0.111	-0.024	-0.077	-0.230	0.068	0.061
20,000	0.447	0.068	-0.041	-0.061	-0.257	0.127	0.013
30,000	0.481	0.036	-0.046	-0.062	-0.257	0.156	-0.001
50,000	0.504	0.009	-0.036	-0.050	-0.252	0.175	-0.019
70,000	0.513	0.004	-0.037	-0.058	-0.263	0.173	-0.013
90,000	0.514	-0.005	-0.037	-0.058	-0.265	0.172	-0.018
120,000	0.517	-0.005	-0.040	-0.051	-0.256	0.169	-0.027
150,000	0.513	-0.004	-0.038	-0.049	-0.257	0.175	-0.021
180,000	0.517	0.002	-0.027	-0.057	-0.266	0.178	-0.020

Table 3.15: Connection Weights at Different Stages of Learning - Model SMB_U_A_03_01_91 ($\eta = 0.02$)

Learn Count	Lag of inputs						
	1	2	3	364	365	366	367
10,000	0.494	0.009	-0.042	-0.052	-0.248	0.178	-0.007
20,000	0.507	-0.009	-0.040	-0.062	-0.268	0.174	-0.009
30,000	0.510	-0.008	-0.040	-0.060	-0.263	0.187	-0.005
40,000	0.503	-0.016	-0.031	-0.050	-0.254	0.184	-0.017
50,000	0.503	-0.013	-0.045	-0.061	-0.255	0.178	-0.021
60,000	0.520	0.001	-0.036	-0.054	-0.264	0.176	-0.025
70,000	0.517	-0.002	-0.026	-0.063	-0.261	0.167	-0.028
80,000	0.508	0.009	-0.031	-0.057	-0.262	0.176	-0.029
90,000	0.529	0.000	-0.032	-0.052	-0.264	0.172	-0.010

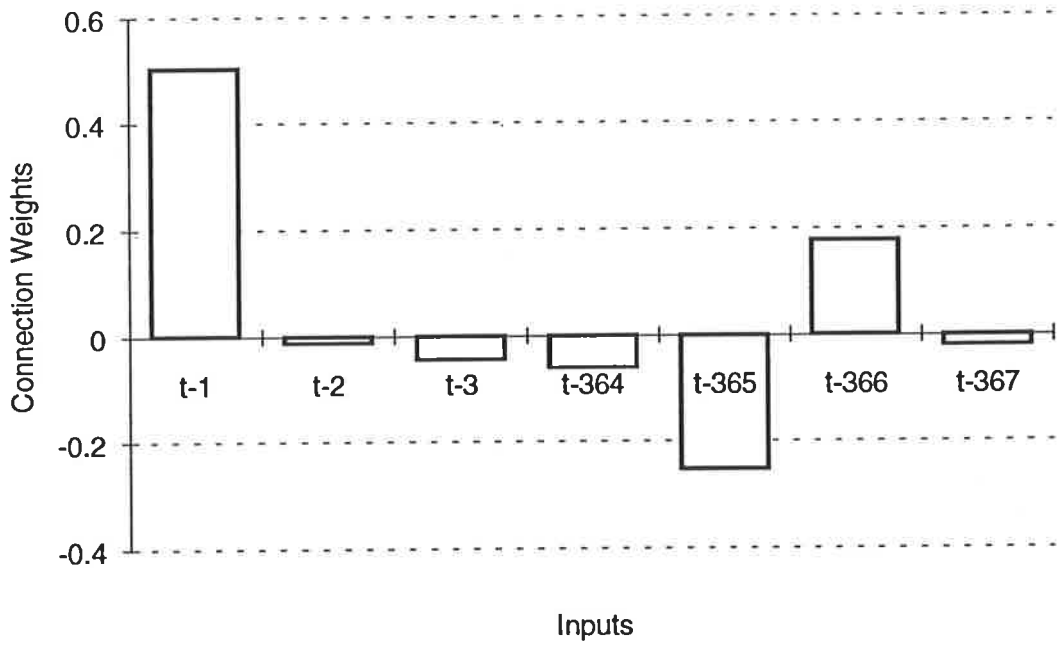


Figure 3.57: Typical Values of Connection Weights

3.4.4.2 One Day Forecasts

The best **differenced** one day forecasts for each network, which are highlighted in Appendix E, were converted to their original (non-differenced) form with the aid of computer program FCAST. The RMSE, AAPE and AAE were then calculated for the **converted** one day forecasts, which are shown in Tables 3.16 to 3.20.

The first network trained (model SMB_U_A_04_02_91) had no hidden layers and a linear transfer function. The results for the best one day forecast obtained are shown in Table 3.16 and Figure 3.58. It can be seen that the forecast is very good and provides a good fit to the historical data, although there is a small amount of noise present.

Table 3.16: Best 1 Day Forecast Obtained Using Model SMB_U_A_04_02_91

Model	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_04_02_91	14.9	1.5	9.6

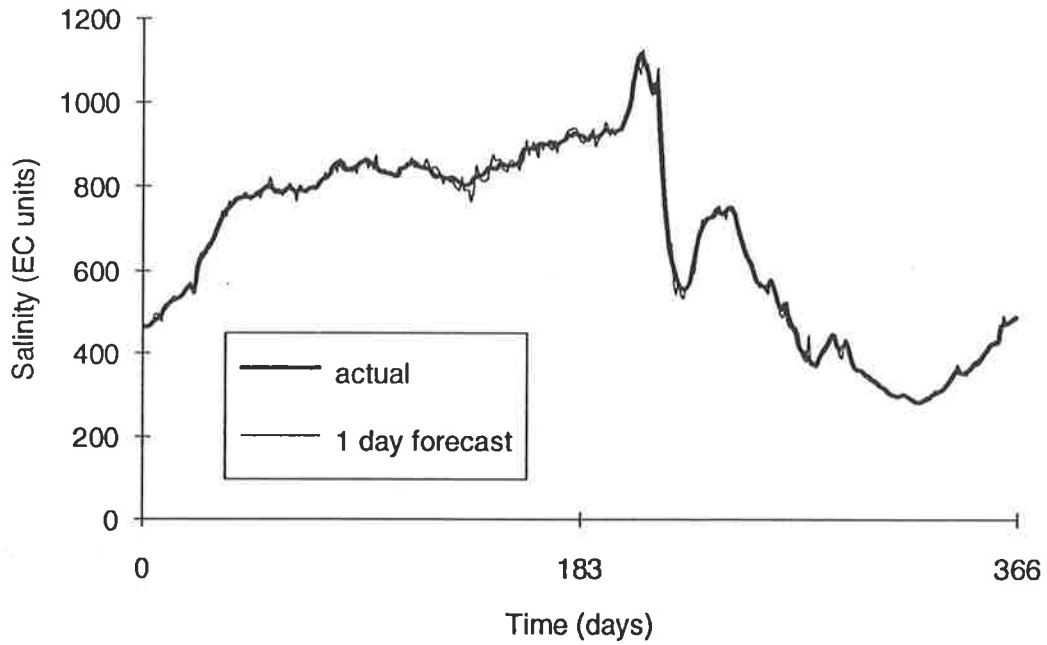


Figure 3.58: Actual and Predicted Salinities Obtained Using Model SMB_U_A_04_02_91 - 1 Day Forecast

Effect of different transfer functions

The effect of using a non-linear transfer function on network performance was investigated by replacing the linear transfer function used in model SMB_U_A_04_02_91 with the hyperbolic tangent transfer function (model SMB_U_A_05_02_91). All other network parameters were identical to those used in model SMB_U_A_04_02_91. As can be seen from Table 3.17, using a non-linear transfer function had no effect on network performance.

Table 3.17: Best 1 Day Forecasts Obtained Using Models SMB_U_A_04_02_91 and SMB_U_A_05_02_91

Model	Transfer Function	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_04_02_91	Linear	14.9	1.5	9.6
SMB_U_A_05_02_91	Tanh	14.9	1.5	9.6

Effect of network geometry

The effect of including one hidden layer was assessed in models SMB_U_A_06_02_91 and SMB_U_A_07_02_91. The network parameters used in model SMB_U_A_06_02_91 were identical to those used in model SMB_U_A_05_02_91, with the exception of the inclusion of six hidden layer nodes. Six hidden layer nodes were chosen, as this is close to the upper limit of $2N^I+1$ nodes suggested by Hecht-Nielsen (1987b), where N^I is the number of inputs (three in this case). The number of hidden layer nodes was decreased to three in model SMB_U_A_07_02_91. As can be seen from Table 3.18, the number of nodes in the first hidden layer had virtually no effect on network performance.

Table 3.18: Best 1 Day Forecasts Obtained Using Models SMB_U_A_05_02_91, SMB_U_A_06_02_91 and SMB_U_A_07_02_91

Model	No. of nodes in hidden layer 1	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_05_02_91	0	14.9	1.5	9.6
SMB_U_A_07_02_91	3	14.9	1.5	9.6
SMB_U_A_06_02_91	6	15.0	1.5	9.6

In model SMB_U_A_08_02_91, a second hidden layer with two nodes was added to the network used in model SMB_U_A_06_02_91. All other network parameters remained unchanged. It should be noted that the ratio of first to second layer hidden nodes used was 3:1, as suggested by Kudrycki (1988). The results in Table 3.19 indicate that adding a second hidden layer had virtually no effect on network performance.

Table 3.19: Best 1 Day Forecasts Obtained Using Models SMB_U_A_06_02_91 and SMB_U_A_08_02_91

Model	No. of nodes in hidden layer 1	No. of nodes in hidden layer 2	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_06_02_91	6	0	15.0	1.5	9.6
SMB_U_A_08_02_91	6	2	14.9	1.5	9.6

Effect of the number of inputs

As described earlier, it was determined that inputs at lags 1, 365 and 366 were clearly dominant for networks that have a linear transfer function and no hidden layers. However, the effect of an increased number of inputs has not yet been determined for a network with hidden layers and a non-linear transfer function. The effect of an increased number of inputs on network performance and the importance of each of the inputs was investigated in model SMB_U_A_09_01_91. Training set 01 was used, which includes inputs at lags 1, 2, 3, 364, 365, 366 and 367. The number of nodes used in the first hidden layer was 15 and the number of nodes used in the second hidden layer was 5. The results obtained were marginally better in terms of RMSE (Table 3.20), but not enough to warrant the inclusion of 4 additional parameters.

Table 3.20: Best 1 Day Forecasts Obtained Using Models SMB_U_A_08_02_91 and SMB_U_A_09_01_91

Model	Number of inputs	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_08_02_91	3	14.9	1.5	9.6
SMB_U_A_09_01_91	7	14.8	1.6	9.6

A sensitivity analysis carried out with the aid of the software's (NeuralWorks Professional II/Plus) "Explain" function (see Section 3.3.4.1) showed that inputs at lags 1, 365 and 366 were again clearly dominant (Figure 3.59).

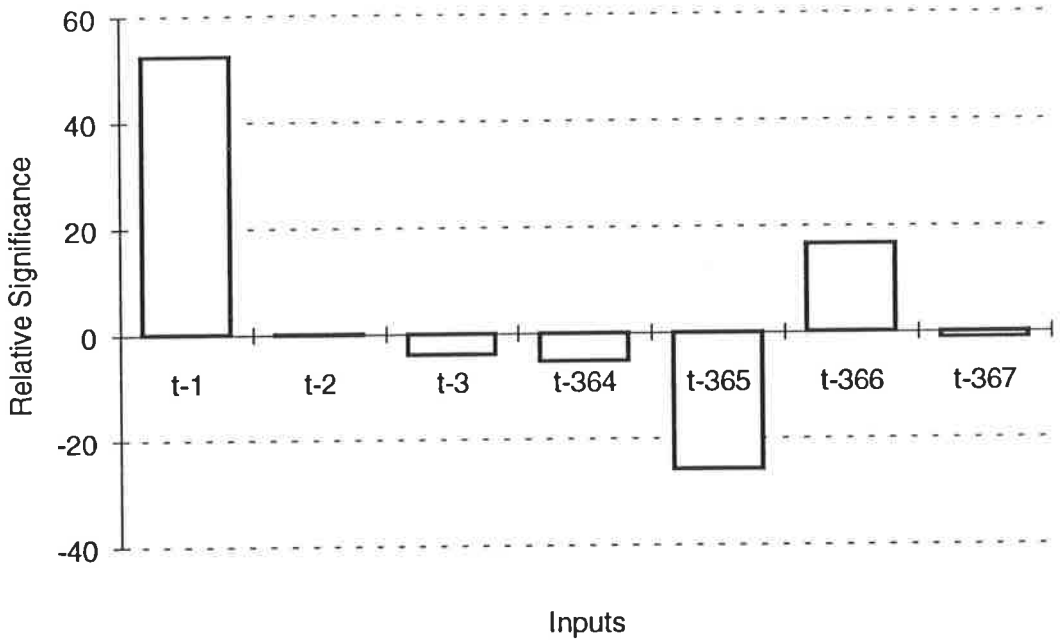


Figure 3.59: Relative Significance of Network Inputs for Model SMB_U_A_09_01_91

3.4.4.3 Fourteen Day Forecasts

The following recursive procedure was used to obtain the 14 day forecasts:

1. Variable g is set equal to 0.
2. The differenced forecast at time t (1 day forecast) is obtained using the ANN model.
3. Variable g is set equal to g + 1.
4. The data set used to obtain the differenced ANN forecasts (the test set) is modified, so that the differenced forecast at time t + g is obtained. If any inputs are required such that (lag of input) > (t-1), the differenced forecasts obtained previously as part of this procedure are used as network inputs.
5. The differenced forecast at time t + g is obtained by presenting the test set obtained in step 4 to the network.
6. Steps 3, 4 and 5 are repeated until g = 13.
7. The differenced forecasts are converted to their non-differenced form with the aid of computer program FCAST (using Equation 3.2).

Initially, a 14 day forecast was obtained using model SMB_U_A_04_02_91, as it gave the best 1 day forecast of the ANN models considered.

The plot of the 14 day forecast obtained using model SMB_U_A_04_02_91 (Figure 3.60) does not exhibit any significant lags. However, it is very noisy. This noise is a

result of the recursive procedure used to obtain the multi-step forecasts. Any noise present in the one day forecast gets amplified as the forecast period increases, as the one day forecast is used to obtain the two day forecast, which in turn is used to obtain the 3 day forecast and so on. This reduces the usefulness of the model, as it is impossible to discern which of the predicted variations in salinity will actually occur. It is interesting to note that the sharp drop in salinity in the vicinity of day 125 can be attributed to a sharp drop in salinity at that time in the previous year (Figure 3.60), which forms part of the forecast as a result of the inputs at lags 365 and 366.

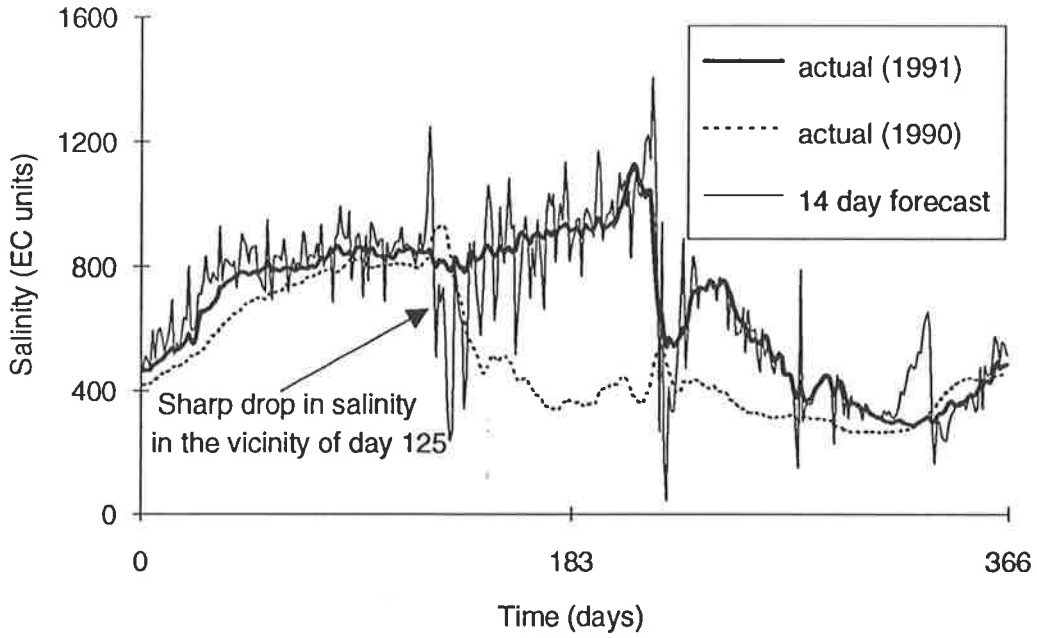


Figure 3.60: Actual and Predicted Salinities Obtained Using Model SMB_U_A_04_02_91 - 14 Day Forecast

Effect of network geometry

A fourteen day forecast was obtained using model SMB_U_A_06_02_91 to assess the effect of hidden layer nodes on longer term forecasts. Model SMB_U_A_06_02_91 was chosen, as it produced the best 1 day forecast for models with hidden layer nodes. It should be noted that model SMB_U_A_09_01_91 was not considered, as it had a greater number of inputs. The results in Table 3.21 show that the ANN model without hidden layer nodes (model SMB_U_A_04_02_91) produced a slightly better 14 day forecast than the model with one hidden layer (model SMB_U_A_06_02_91). This can be attributed to the better starting point (1 day forecast) of model SMB_U_A_04_02_91. Consequently, it can be concluded that the addition of hidden layer nodes has no impact on longer term forecasts when using differenced data.

Table 3.21: Best 14 Day Forecasts Obtained Using Models SMB_U_A_04_02_91 and SMB_U_A_06_02_91

Model	No. of nodes in hidden layer 1	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_04_02_91	0	123.8	13.3	80.9
SMB_U_A_06_02_91	6	125.2	13.6	80.6

3.4.4.4 Summary

The results obtained using the UANN models trained with the differenced data indicate that the one day forecasts obtained were not significantly affected by:

1. The transfer function.
2. The network geometry (i.e. the number of hidden layers and the number of nodes per hidden layer).
3. The number of inputs, provided the dominant inputs are included.

The results obtained also indicate that the presence of hidden layer nodes does not appear to have any effect on the longer term forecasts produced when using differenced data and the recursive forecasting procedure.

3.4.5 Development of Models Using the Raw Data

Using the raw data to train and test neural network models is much quicker than using the differenced data, as network inputs do not have to be differenced before training commences and the forecasts obtained do not have to be converted back to their non-differenced form.

A number of models were developed to investigate the effect of network geometry, the number of inputs and the number of outputs on network performance. A learning rate (η) of 0.1, a momentum value (μ) of 0.6, an epoch size (ε) of 5 and the hyperbolic tangent transfer function were used unless stated otherwise. Details of the network geometry and the internal parameters used for the various models are given in Appendix D. Details of the training / testing sets used for the various models are given in Appendix C. The number of training samples presented to each network was 100,000 and the testing interval used was 5,000 unless stated otherwise. The performance of the models at various stages of learning is given in Appendix E.

Forecasts were obtained for forecasting periods of one and 14 days. It should be noted that one more year of data was available when training the network with the raw data than when training the network with the differenced data, as, in the latter case, one year of data (1987) is lost as a result of the differencing process.

3.4.5.1 Determination of Initial Model Inputs

When using the raw data, the ACF and the PACF can no longer be used as a reliable tool for determining the initial model inputs. However, it is reasonable to assume that the inputs at low lags and the inputs in the vicinity of one seasonal lag are still dominant. In order to ensure that all significant inputs were included, and keeping in mind that the results obtained in Section 3.4.4 suggest that unnecessary inputs do not have a significant effect on model performance, training / testing set 03 was used initially, which consists of 41 inputs at lags 1, 2, ..., 19, 20, 354, 355, ..., 375 and 376 and 3 outputs at lags 0, -7 and -13. It should be noted that the 1 and 14 day forecasts were obtained simultaneously when the networks were tested, corresponding to the outputs at lags 0 and -13.

3.4.5.2 One Day Forecasts

Initially, a model without any hidden layers was developed (model SMB_U_A_10_03_91), so that a direct comparison could be obtained between a model that is trained using the raw data and a model that is trained using the differenced data. A learning rate of 0.02 and a testing interval of 5,000 were used.

As can be seen from Table 3.22, the one day forecast obtained when the differenced data were used (model SMB_U_A_04_02_91) was significantly better than that obtained when the raw data were used (model SMB_U_A_10_03_91). As discussed above, neural networks without hidden layers operate in a manner similar to univariate time series models. Consequently, one would expect to obtain better one day forecasts when the time series data used to train the network are stationary.

Table 3.22: Best 1 Day Forecasts Obtained Using Models SMB_U_A_04_02_91 and SMB_U_A_10_03_91

Model	Data	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_04_02_91	Differenced	14.9	1.5	9.6
SMB_U_A_10_03_91	Raw	17.1	2.3	12.2

In order to assess the importance of each of the network inputs, the connection weights obtained for model SMB_U_A_10_03_91 were examined (Figure 3.61). It is interesting to note that the inputs that were found to be significant were quite different when using the differenced data (lags 1, 365 and 366 - Figure 3.57) compared to when using the raw data (lags 1, 2 and 5). The most notable difference is the significance of inputs in the vicinity of one seasonal lag when using the differenced data, which is not the case when using the raw data.

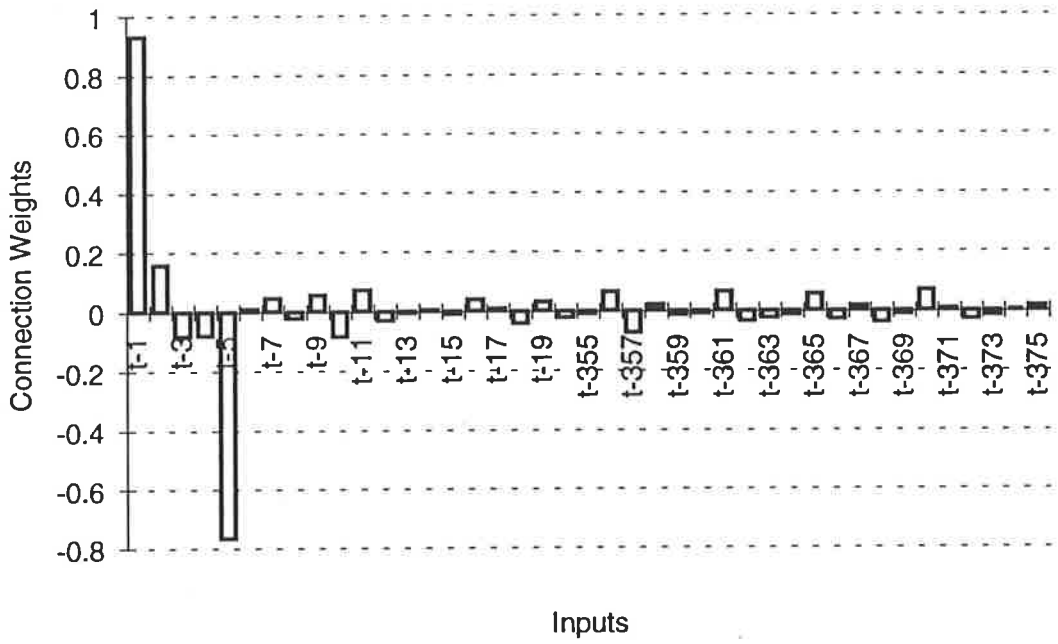


Figure 3.61: Connection Weights for Model SMB_U_A_10_03_91

Effect of network geometry

The effect of the addition of hidden layer nodes on network performance was investigated in models SMB_U_A_11_03_91 and SMB_U_A_12_03_91. The number of nodes used in the first hidden layer was 45 for model SMB_U_A_11_03_91 and 60 for model SMB_U_A_12_03_91. The number of nodes used in the second hidden layer was 15 for model SMB_U_A_11_03_91 and 20 for model SMB_U_A_12_03_91.

The best results obtained using the above models are shown in Table 3.23. As can be seen, the network with no hidden layers produced a considerably better one day forecast. The networks with two layers of hidden nodes produced very similar forecasts, suggesting that the number of hidden layer nodes does not have a pronounced effect on network performance; only the absence or presence of hidden layers.

Table 3.23: Best 1 Day Forecasts Obtained Using Models SMB_U_A_10_03_91, SMB_U_A_11_03_91 and SMB_U_A_12_03_91

Model	No. of nodes in hidden layer 1	No. of nodes in hidden layer 2	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_10_03_91	0	0	17.1	2.3	12.2
SMB_U_A_11_03_91	45	15	25.0	2.9	16.5
SMB_U_A_12_03_91	60	20	24.4	2.7	15.2

In order to assess the importance of each of the inputs for the models with hidden layers, a sensitivity analysis was carried out using the software's (NeuralWorks Professional II/Plus) "Explain" function for model SMB_U_A_11_03_91. This does not enable a direct comparison to be carried out between models SMB_U_A_11_03_91 and SMB_U_A_10_03_91, but enables the relative importance of each of the inputs to be assessed. Figures 3.62 and 3.61 show that the inputs at lags 1 and 2 were dominant for model SMB_U_A_11_03_91, whereas the inputs at lags 1, 2 and 5 were dominant for model SMB_U_A_10_03_91. This accounts for the difference in performance of the models and indicates that the mechanism by which the forecasts are obtained is different when hidden layer nodes are present. In the absence of any hidden layers, the network output is a direct combination of network inputs. This is not the case when hidden layers are present. The presence of hidden layers results in a much more complex relationship between network inputs and outputs. The hidden layers enable the network to determine some of the underlying relationships present in non-stationary data (e.g. trends, seasonality). As discussed in Section 3.4.4.2, when the differenced data were used, the addition of hidden layers had no effect. This is due to the fact that after differencing, all underlying relationships in the data had already been removed, so that there is no benefit in using hidden layers. However, as indicated by the results in Table 3.23, when underlying relationships are present in the data, the presence of hidden layers does have an effect. It should be noted, though, that in the case study considered here, the addition of the hidden layer nodes resulted in a worse one day forecast. The effect of the presence of hidden layer nodes on longer term forecasting is discussed in Section 3.4.5.3.

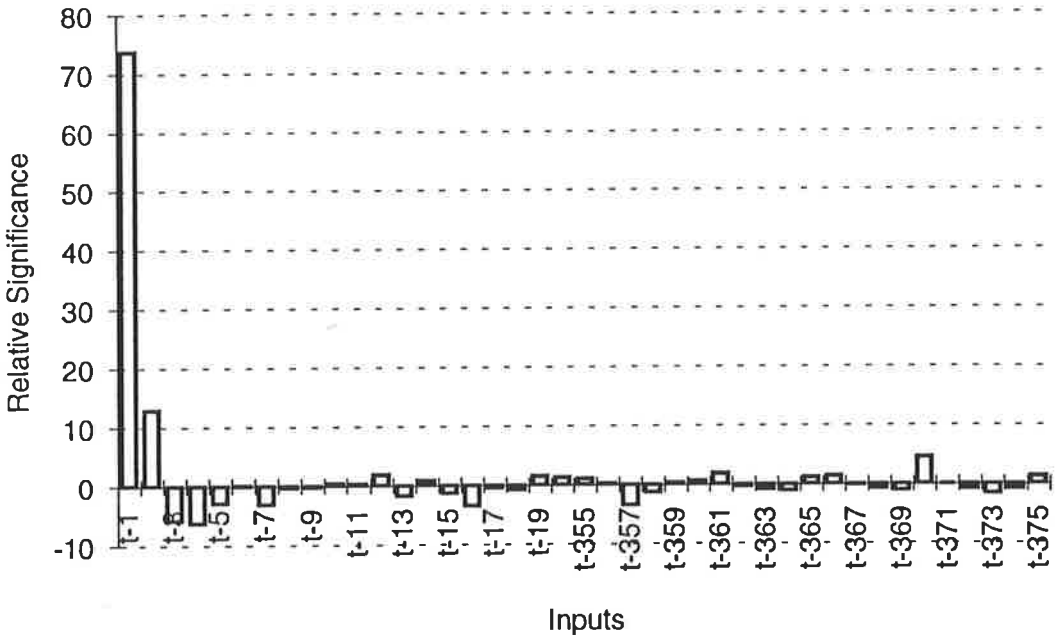


Figure 3.62: Relative Significance of Network Inputs for Model SMB_U_A_11_03_91

The plots of the one day forecasts produced by models SMB_U_A_10_03_91 and SMB_U_A_12_03_91 are shown in Figures 3.63 and 3.64 respectively. It can be seen that model SMB_U_A_10_03_91 models the sharp variations in salinity better than model SMB_U_A_12_03_91, especially the sharp peak in the vicinity of day 200.

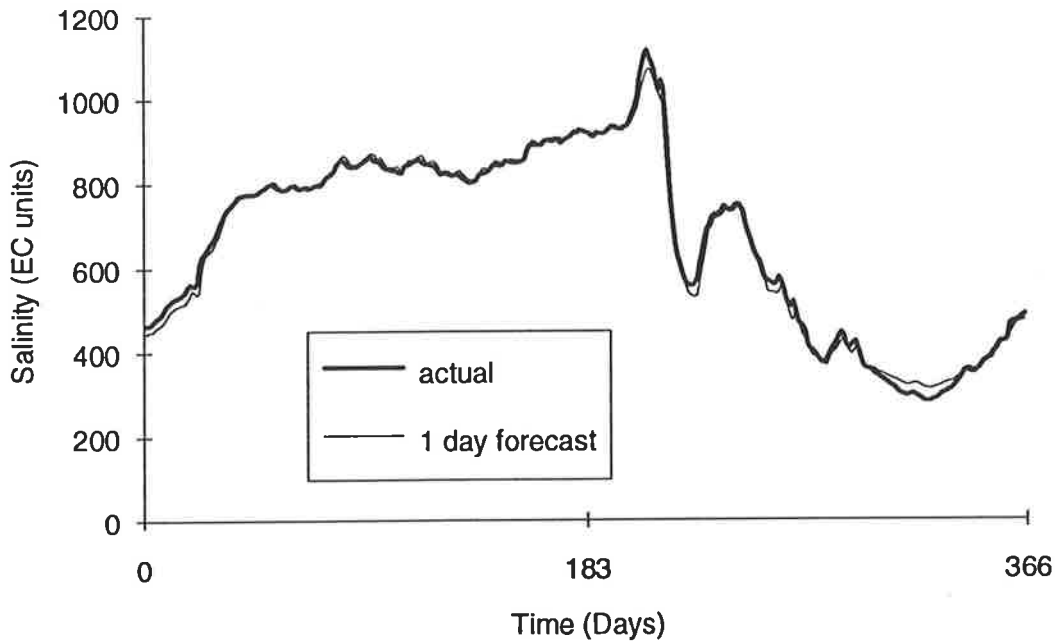


Figure 3.63: Actual and Predicted Salinities Obtained Using Model SMB_U_A_10_03_91 - 1 Day Forecast - No Hidden Layers

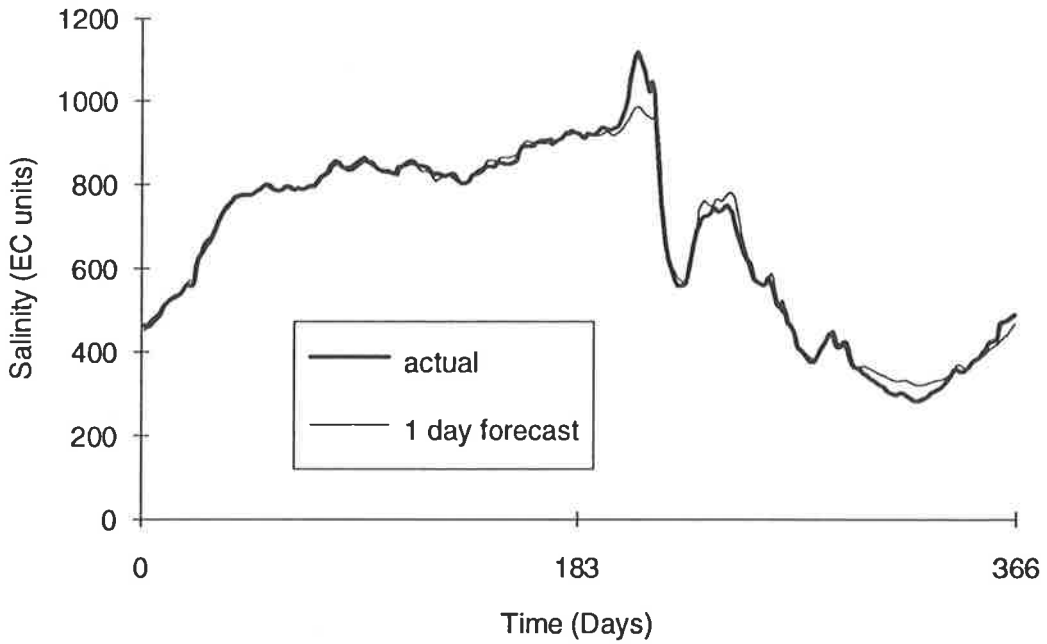


Figure 3.64: Actual and Predicted Salinities Obtained Using Model SMB_U_A_12_03_91 - 1 Day Forecast - Two Hidden Layers

Effect of the number of inputs

The effect of decreasing the number of inputs on network performance was assessed in models SMB_U_A_12_03_91 and SMB_U_A_13_04_91. The details of model SMB_U_A_12_03_91 were discussed previously. Training / testing set 04 was used for model SMB_U_A_13_04_91, which consists of 15 inputs at lags 1, 2, ..., 14, and 15 and 3 outputs at lags 0, -7 and -13. The number of nodes used in the first hidden layer was 24 and the number of nodes used in the second hidden layer was 8. All other network parameters were the same as in model SMB_U_A_12_03_91.

Decreasing the number of inputs from 41 to 15 resulted in a slight reduction in network performance, as shown in Table 3.24. This suggests that network performance is not influenced greatly by the number of inputs, provided the dominant inputs are included.

Table 3.24: Best 1 Day Forecasts Obtained Using Models SMB_U_A_12_03_91 and SMB_U_A_13_04_91

Model	No. of inputs	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_12_03_91	41	24.4	2.7	15.2
SMB_U_A_13_04_91	15	25.1	2.7	15.7

3.4.5.3 Fourteen Day Forecasts

Two methods were used to obtain the 14 day forecasts; the direct method and the recursive method.

A. Direct method

All networks using the raw data were trained to produce 14 day forecasts, as indicated by the inclusion of outputs at lag -13 in all training / testing sets used. This is by far the simplest and quickest method of obtaining 14 day forecasts, as there is no need for recursive procedures or data conversions.

Initially, a 14 day forecast was obtained using model SMB_U_A_10_03_91, as it gave the best 1 day forecast of the models considered. A description of the network parameters and training / testing set used is given in Section 3.4.5.2.

The results in Table 3.25 show that the 14 day forecast obtained from the model using the direct forecasting method and the raw data (model SMB_U_A_10_03_91) is significantly better than the 14 day forecast obtained from the model using the recursive forecasting method and the differenced data (model SMB_U_A_04_02_91).

Table 3.25: Best 14 Day Forecasts Obtained Using Models SMB_U_A_10_03_91 (direct) and SMB_U_A_04_02_91 (recursive)

Model	Forecasting method	Data	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_10_03_91	Direct	Raw	105.1	12.1	74.5
SMB_U_A_04_02_91	Recursive	Differenced	123.8	13.3	80.9

The plot of the 14 day forecast obtained using model SMB_U_A_10_03_91 is shown in Figure 3.65. A comparison of Figures 3.65 and 3.60 shows that the forecasts produced using the differenced data and the forecasts obtained using the raw data have markedly different characteristics. The forecasts produced using the raw data are very smooth, but tend to exhibit a pronounced shift. The forecasts produced by the models trained with the differenced data, on the other hand, are very noisy, as a result of the combination of differencing and the recursive procedure used to obtain the longer term forecasts. However, they do not exhibit a pronounced shift.

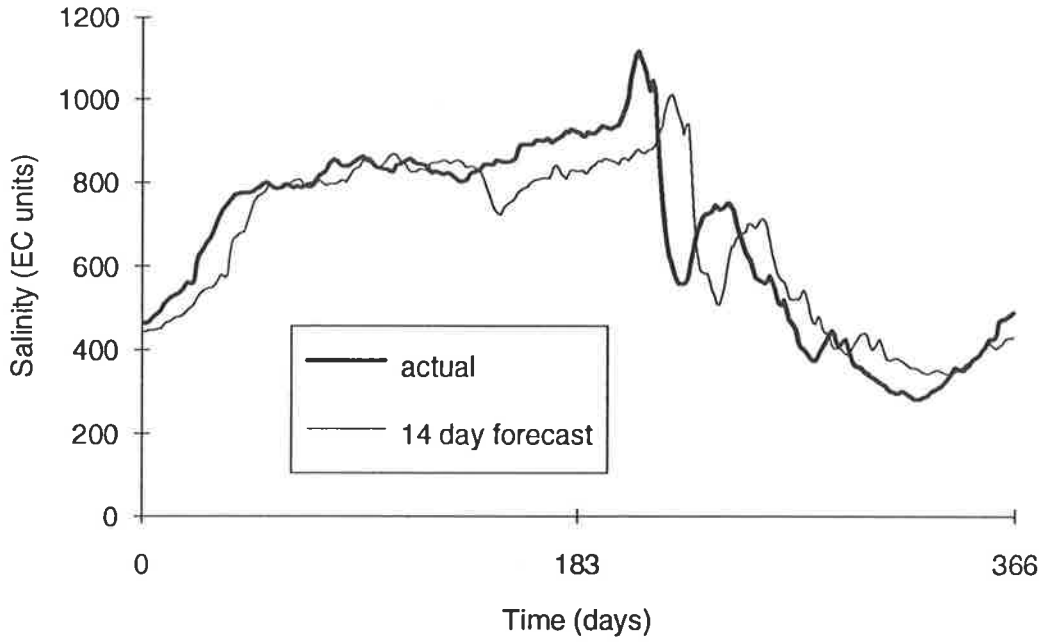


Figure 3.65: Actual and Predicted Salinities Obtained Using Model SMB_U_A_10_03_91 - 14 Day Forecast (Direct Method)

Effect of network geometry

The effect of the addition of hidden layer nodes on network performance was assessed using models SMB_U_A_11_03_91 and SMB_U_A_12_03_91. A description of the network parameters and training / testing set used for these models is given in Section 3.4.5.2. As described earlier, the testing interval used was 5,000. Additional testing was carried out at intervals of 1,500 training samples from learn counts 10,000 to 25,000 for model SMB_U_A_12_03_91.

The best results obtained for models SMB_U_A_10_03_91, SMB_U_A_11_03_91 and SMB_U_A_12_03_91 are shown in Table 3.26. The results show quite a large difference in performance between models. The model without hidden layer nodes performed worst, while the model with the largest number of hidden layer nodes performed best. This is in contrast to the results obtained for the one day forecasts (Table 3.23). The different mechanisms by which the models obtain their outputs discussed in Section 3.4.5.2 are responsible for the difference in results. Models SMB_U_A_11_03_91 and SMB_U_A_12_03_91 are able to determine some of the underlying relationships in the data with the aid of their hidden layer nodes and are hence better equipped to produce longer term forecasts. An increased number of hidden layer nodes also had a positive effect on network performance.

Table 3.26: Best 14 Day Forecasts Obtained Using Models SMB_U_A_10_03_91, SMB_U_A_11_03_91 and SMB_U_A_12_03_91 (Direct Method)

Model	No. of nodes in hidden layer 1	No. of nodes in hidden layer 2	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_10_03_91	0	0	105.1	12.1	74.5
SMB_U_A_11_03_91	45	15	93.9	11.1	70.2
SMB_U_A_12_03_91	60	20	87.9	10.9	64.6

The relative significance of each model input in producing the 14 day forecast was assessed for models SMB_U_A_10_03_91 and SMB_U_A_11_03_91. This was done by obtaining the connection weights for model SMB_U_A_10_03_91 and carrying out a sensitivity analysis for model SMB_U_A_11_03_91 with the aid of the software's (NeuralWorks Professional II/Plus) "Explain" function. Figures 3.66 and 3.67 show that the relative significance of each of the inputs is different when the networks are trained to produce a 1 day forecast compared to when they are trained to produce a 14 day forecast. This emphasises one of the advantages of neural networks, in that they can be trained to produce a specific n-step forecast, rather than being trained to produce a one-step forecast and then rely on a recursive procedure to obtain longer term forecasts. By training networks to produce particular n-step forecasts, they are able to determine relationships specific to that forecasting period.

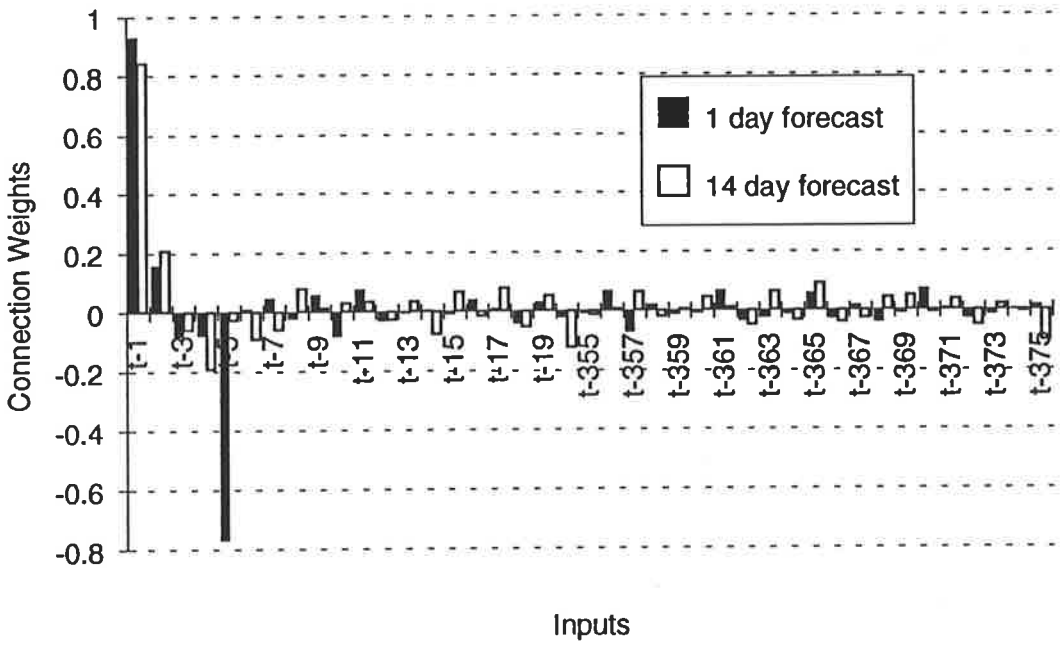


Figure 3.66: Comparison of Connection Weights for 1 and 14 Day Forecasts for Model SMB_U_A_10_03_91 (No Hidden Layers)

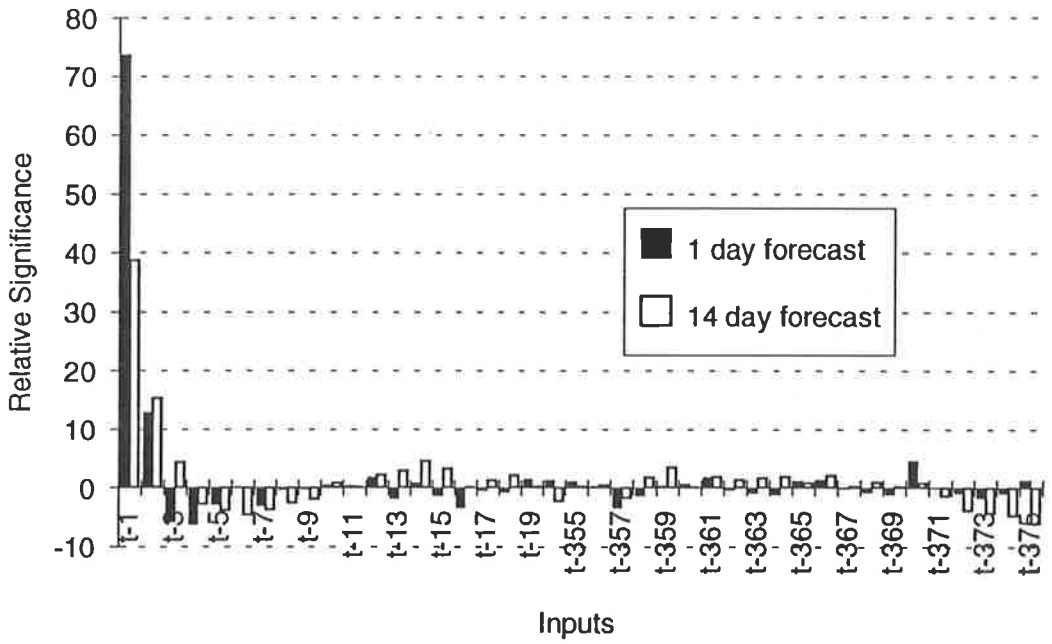


Figure 3.67: Comparison of Relative Significance of Inputs for 1 and 14 Day Forecasts for Model SMB_U_A_11_03_91 (Two Hidden Layers)

Plots of the 14 day forecasts produced by models SMB_U_A_10_03_91 and SMB_U_A_12_03_91 are shown in Figures 3.65 and 3.68 respectively. The plots show that both 14 day forecasts exhibit a pronounced shift. The shift is less than 14 days for most periods. However, there appears to be a shift of approximately 14 days in the region where major variations in salinity occur; between days 200 and 320. Consequently, the models are not very useful for predicting sharp rises and falls in salinity. However, the overall performance of the model is far better than that of a naive model (i.e. $z_{t+14} = z_t$), as shown in Table 3.27.

Table 3.27: Best 14 Day Forecasts Obtained Using Model SMB_U_A_12_03_91 (Direct Method) and a Naive Model

Model	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_12_03_91	87.9	10.9	64.6
Naive	117.0	12.5	76.7

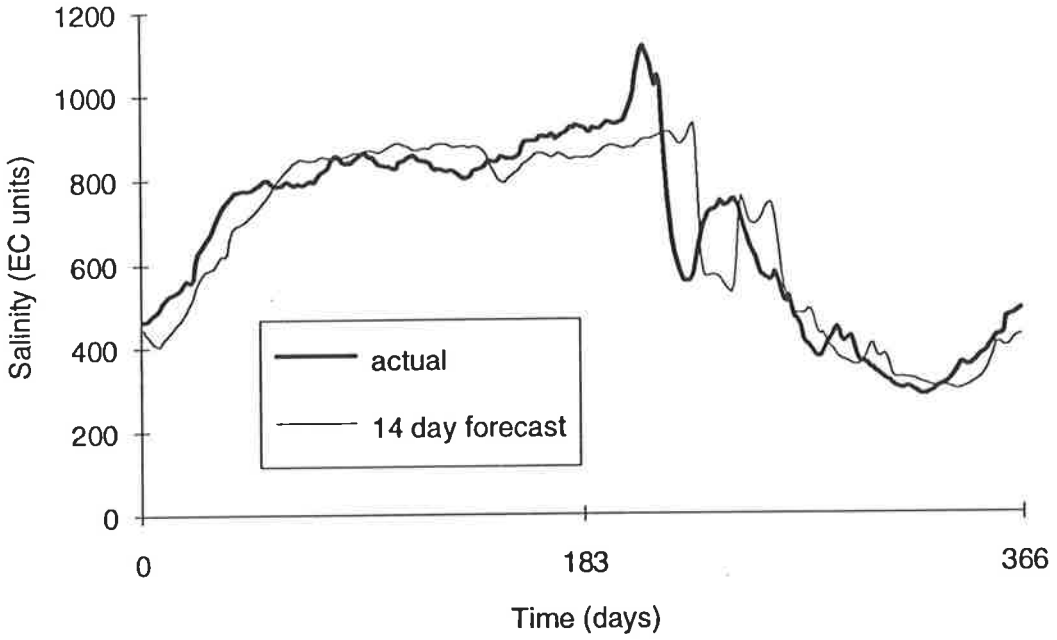


Figure 3.68: Actual and Predicted Salinities Obtained Using Model SMB_U_A_12_03_91 - 14 Day Forecast (Direct Method)

Effect of the number of outputs

The effect of the number of outputs on network performance was assessed using models SMB_U_A_11_03_91 and SMB_U_A_14_05_91. The network parameters used in model SMB_U_A_14_05_91 were identical to those used in model SMB_U_A_11_03_91. Training / testing set 05 was used for model SMB_U_A_14_05_91, which is identical to training / testing set 03, with the exception that there is only one output at lag -13.

As can be seen from Table 3.28, decreasing the number of outputs from 3 to 1 did not have a significant effect on network performance.

Table 3.28: Best 14 Day Forecasts Obtained Using Models SMB_U_A_11_03_91 and SMB_U_A_14_05_91 (Direct Method)

Model	No. of outputs	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_11_03_91	3	93.9	11.1	70.2
SMB_U_A_14_05_91	1	95.7	11.7	73.5

Effect of the number of inputs

The effect of decreasing the number of inputs on network performance was assessed in models SMB_U_A_12_03_91 and SMB_U_A_13_04_91. The network parameters and training / testing sets used for models SMB_U_A_12_03_91 and SMB_U_A_13_04_91 have been described earlier.

The results in Table 3.29 show that reducing the number of inputs from 41 to 15 caused a slight decrease in network performance.

Table 3.29: Best 14 Day Forecasts Obtained Using Models SMB_U_A_12_03_91 and SMB_U_A_13_04_91 (Direct Method)

Model	No. of inputs	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_12_03_91	41	87.9	10.9	64.6
SMB_U_A_13_04_91	15	91.5	10.2	60.2

B. Recursive method

A recursive procedure, similar to that outlined in Section 3.4.4.3, was used to obtain the 14 day forecasts. As the raw data were used to train and test the models, the forecasts obtained were already in their non-differenced form, so that step 7 in the procedure described in Section 3.4.4.3 was not required.

Initially, a 14 day forecast was obtained using model SMB_U_A_10_03_91, as it gave the best 1 day forecast of the models trained with the raw data. The details of model SMB_U_A_10_03_91 are given in Section 3.4.5.2. The results in Table 3.30 show that the 14 day forecast obtained using model SMB_U_A_10_03_91 was far worse than that obtained using model SMB_U_A_04_02_91. As was the case with the previous recursive forecasts obtained, the model producing the worst one day forecast produced the worst 14 day forecast. This is what one would expect, as the 1 day forecasts are used to obtain the 2 day forecasts, which are in turn used to obtain the 3 day forecasts and so on. This progression of errors is clearly demonstrated in Figures 3.69, 3.70 and 3.71, which show the 3, 5 and 14 day recursive forecasts obtained using model SMB_U_A_10_03_91. The plots show that the longer the forecasting period, the greater the shift in the forecast. In addition, detailed variations in salinity are lost as the forecasting period increases.

Table 3.30: Comparison of the Best 14 Day Forecasts Obtained Using Models SMB_U_A_04_02_91 and SMB_U_A_10_03_91 (Recursive Method)

Model	Data	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_04_02_91	Differenced	123.8	13.3	80.9
SMB_U_A_10_03_91	Raw	159.0	18.7	120.2

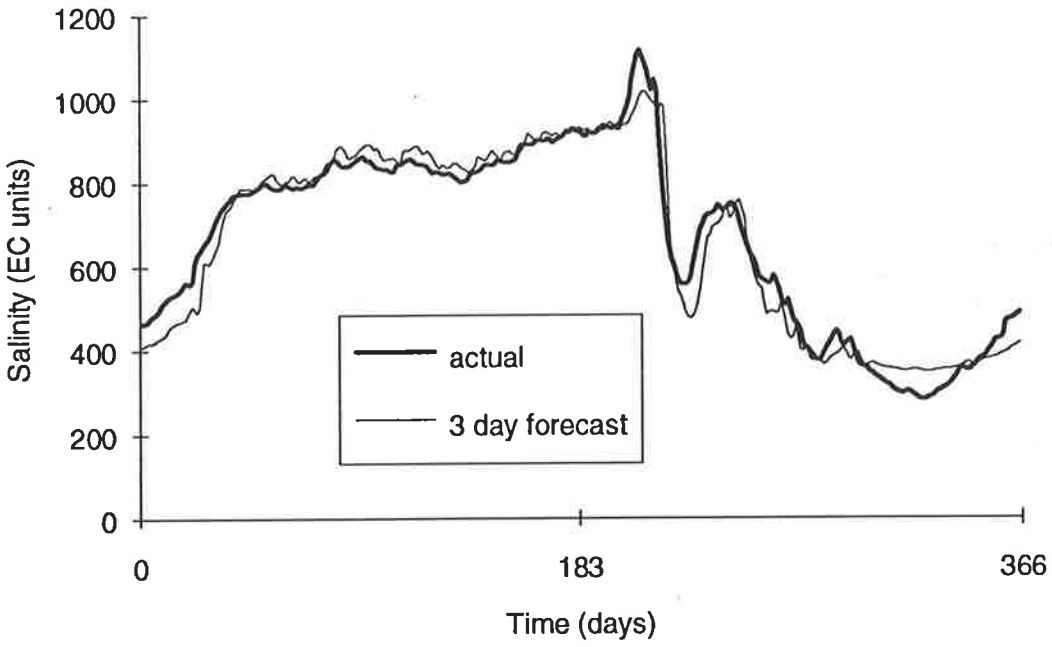


Figure 3.69: Actual and Predicted Salinities Obtained Using Model SMB_U_A_10_03_91 - 3 Day Forecast (Recursive Method)

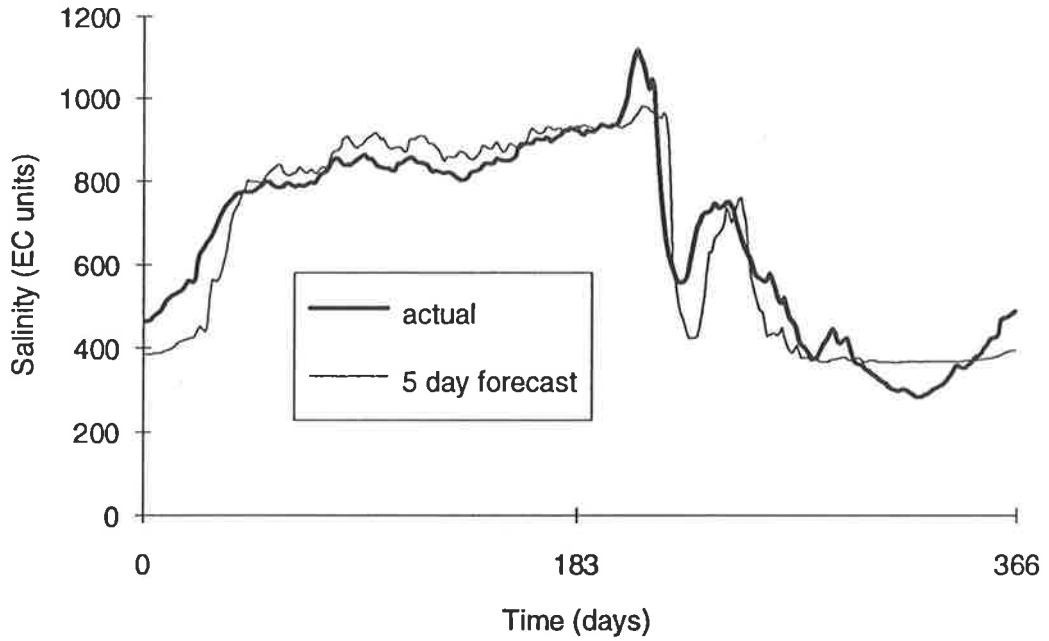


Figure 3.70: Actual and Predicted Salinities Obtained Using Model SMB_U_A_10_03_91 - 5 Day Forecast (Recursive Method)

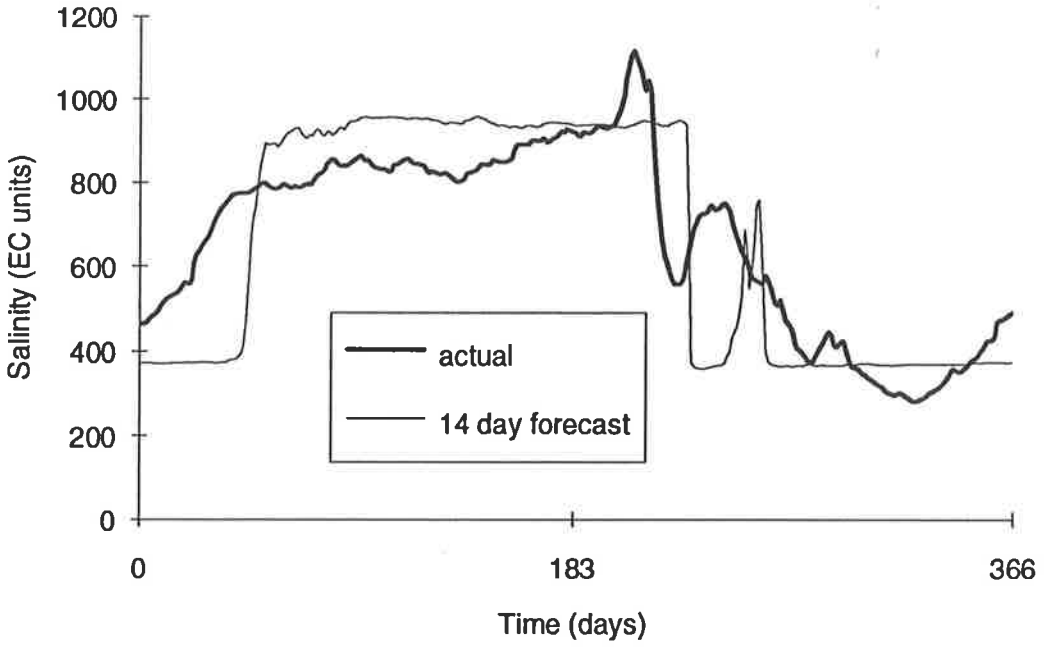


Figure 3.71: Actual and Predicted Salinities Obtained Using Model SMB_U_A_10_03_91 - 14 Day Forecast (Recursive Method)

Effect of network geometry

A 14 day forecast was obtained using model SMB_U_A_12_03_91 to assess the effect of hidden layer nodes on longer term forecasts. Model SMB_U_A_12_03_91 was chosen, as it produced the best 1 day forecast of the models with hidden layer nodes. The details of model SMB_U_A_12_03_91 are given in Section 3.4.5.2. The results in Table 3.31 show that the model without hidden layer nodes (model SMB_U_A_10_03_91) produced a better 14 day forecast than the model with two hidden layers (model SMB_U_A_12_03_91). Again, this can be attributed to the better one day forecast produced by model SMB_U_A_10_03_91, suggesting that the presence of hidden layer nodes does not affect the recursive forecasting procedure. Inspection of the plots of the 14 day forecasts for models SMB_U_A_10_03_91 and SMB_U_A_12_03_91 (Figures 3.71 and 3.72) shows that they vary considerably. The plots indicate that model SMB_U_A_12_03_91 is able to model salinity better than model SMB_U_A_10_03_91 when only small variations are occurring. This might be due to the presence of hidden layer nodes in model SMB_U_A_12_03_91. The plots also show that model SMB_U_A_12_03_91 places greater emphasis on inputs at one seasonal lag, as indicated by the sharp drop in the predicted salinity at approximately 130 days. As explained previously, this is due to a sharp drop in salinity at that time in the previous year (Figure 3.60).

Table 3.31: Best 14 Day Forecasts Obtained Using Models SMB_U_A_10_03_91 and SMB_U_A_12_03_91 (Recursive Method)

Model	No. of Hidden Layers	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_10_03_91	0	159.0	18.7	120.2
SMB_U_A_12_03_91	2	169.5	25.0	122.6

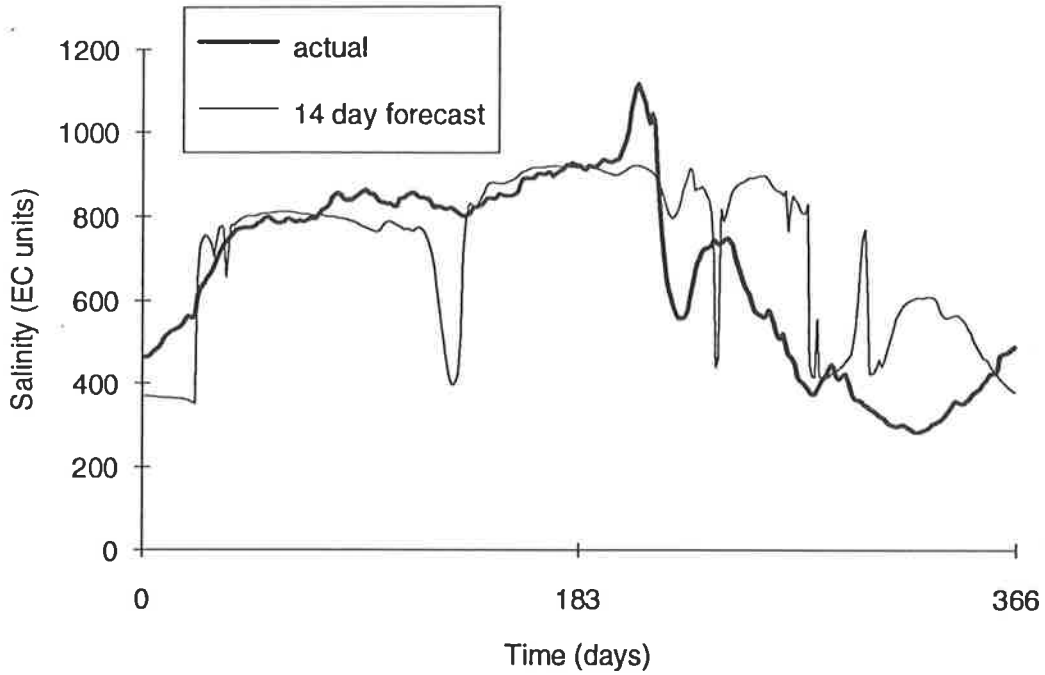


Figure 3.72: Actual and Predicted Salinities Obtained Using Model SMB_U_A_12_03_91 - 14 Day Forecast (Recursive Method)

Effect of the number of inputs

A 14 day forecast was obtained using model SMB_U_A_13_04_91 to assess the effect of a reduced number of inputs on longer term forecasts. The details of model SMB_U_A_13_04_91 are given in Section 3.4.5.2. The results in Table 3.32 show that the model with the greater number of inputs (model SMB_U_A_10_03_91) produced a better 14 day forecast. Again, this can be attributed to the better one day forecast produced by model SMB_U_A_10_03_91. Inspection of the plots of the 14 day forecasts for models SMB_U_A_10_03_91 and SMB_U_A_13_04_91 (Figures 3.71 and 3.73) shows that they vary considerably. Figure 3.73 shows that for model SMB_U_A_13_04_91, the predicted salinity drops off very early and that there is no sharp drop in salinity in the vicinity of day 130, as there are no network inputs at seasonal lags.

Table 3.32: Best 14 Day Forecasts Obtained Using Models SMB_U_A_10_03_91 and SMB_U_A_13_04_91 (Recursive Method)

Model	No. of Inputs	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_A_10_03_91	41	169.5	25.0	122.6
SMB_U_A_13_04_91	15	179.0	17.6	118.1

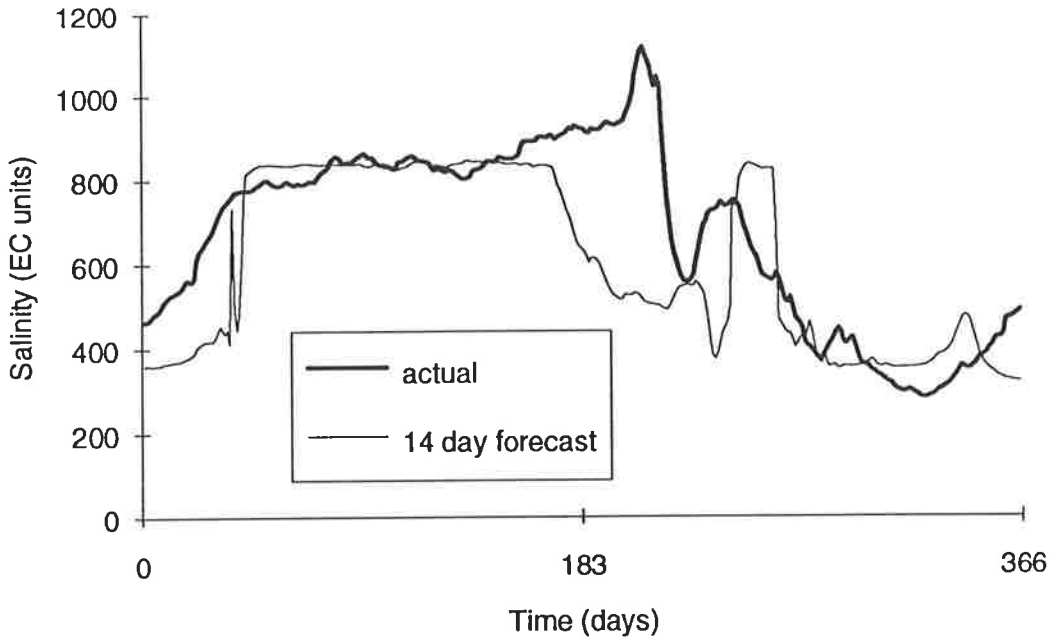


Figure 3.73: Actual and Predicted Salinities Obtained Using Model SMB_U_A_13_04_91 - 14 Day Forecast (Recursive Method)

3.4.5.4 Summary

The results obtained using the univariate ANN models trained with the raw data indicate that:

1. The models without hidden layers provide significantly better one day forecasts than the models with hidden layers.
2. For networks with hidden layers, the number of hidden layer nodes does not have a significant effect on model performance for one day forecasts.
3. Model performance increases with an increase in the number of hidden layer nodes for 14 day forecasts.
4. The number of inputs has no significant impact on model performance, provided the dominant inputs are included.
5. The number of outputs has no significant impact on model performance.

3.4.6 Results / Discussion

The best results obtained using the various forecasting methods discussed are shown in Tables 3.33 and 3.34 for the 1 and 14 day forecasts respectively. It should be noted that in Tables 3.33 and 3.34, the only model which includes hidden layers is the one using raw data and the direct method of forecasting. The performance of the UANN models was compared with that of a naive forecasting model. In the naive model used here, the

forecast value was taken to be the latest actual value (i.e. $z_t = z_{t-g}$, where g = 1 for the one day forecast and g = 14 for the 14 day forecast).

Table 3.33: Best One Day Forecasts Obtained Using the Various Forecasting Methods

	RMSE (EC units)	AAPE (%)	AAE (EC units)
ANN (differenced)	14.8	1.6	9.6
ANN (raw)	17.1	2.3	12.2
Naive	13.8	1.3	7.8

Table 3.34: Best 14 Day Forecasts Obtained Using the Various Forecasting Methods

	RMSE (EC units)	AAPE (%)	AAE (EC units)
ANN (differenced)	123.8	13.3	80.9
ANN (raw - recursive)	159.0	18.7	120.2
ANN (raw - direct)	87.9	10.9	64.6
Naive	117.0	12.5	76.7

The results obtained indicate that the model using the differenced data produced a better one day forecast than the one using the raw data. However, both models did not perform as well as the naive model, suggesting that the UANN models are not useful for producing one day forecasts of salinity in the River Murray at Murray Bridge.

The best 14 day forecast was produced by the model trained with the raw data, followed by the naive model, the model trained with the differenced data and the model trained with the raw data using the recursive forecasting procedure. These results indicate that the model using the raw data in conjunction with the direct forecasting method is a useful tool for forecasting salinity in the River Murray at Murray Bridge 14 days in advance, as it performed significantly better than the naive model. At the same time, the 14 day forecasts obtained using the recursive procedure were not useful.

It should be noted that when the recursive forecasting method was used, the model that produced the best one day forecast also produced the best 14 day forecast. The longer term forecasts obtained were not affected by the type of data used (i.e. differenced or raw) or the number of hidden layers.

The number of model inputs used did not have a significant effect on model performance, provided the dominant inputs were included. The inclusion of less significant, inputs did not introduce any model noise, but resulted in a slight improvement in model performance. This is in agreement with the results obtained by Tang et al. (1991).

The results obtained indicate that the hidden layer nodes perform a task similar to differencing. When the models were trained with the differenced data, the addition of hidden layer nodes had no effect on model performance, as all the underlying relationships in the data (i.e. trends, seasonal variations) had already been removed. However, when the models were trained with the raw data, the addition of hidden layer nodes affected the performance of the models, suggesting that the hidden layer nodes detect some of the underlying relationships present in the data.

The best 14 day forecast was obtained when the direct, rather than the recursive, forecasting procedure was used. This is in contrast to the results obtained by Lapedes and Farber (1988), who found that the recursive procedure produced better results in predicting future values of a chaotic time series. This suggests that forecasting accuracy not only depends on the forecasting method used, but also on the nature of the time series.

Using the differenced data resulted in better one day forecasts. However, the best 14 day forecast was obtained when the raw data were used in conjunction with two hidden layers of nodes and the direct forecasting procedure. As the one day forecasts obtained were worse than that obtained using the naive model, and the focus in this research is on longer term forecasting, it was decided to use the raw data together with hidden layer nodes and the direct method of forecasting in the development of the multivariate ANN model. This has the additional advantages that there is no need for differencing and undifferencing the data, and that it is not necessary to use a recursive procedure to obtain multi-step forecasts.

3.4.7 Real Time Forecasts

In this section, real time 1, 5 and 14 day forecasts were obtained for 1991 using the raw data and the direct method of forecasting (model SMB_U_A_15_06_91). When obtaining real time forecasts, the testing set cannot be used to assess the generalisation ability of the model at various stages of learning in order to optimise model performance. Instead, a fixed number of training samples have to be presented to the network before the forecasts are obtained.

It was decided to use inputs at lags 1, 2, ..., 15 (training / testing set 06), as the tests carried out in Section 3.4.5 indicated that reducing the number of inputs from 41 to 15 does not have an adverse effect on model performance, and training speed is increased by using smaller network geometries. The learning rate used was 0.05 and the number of training samples presented to the network was 50,000. The results obtained using model SMB_U_A_15_06_91 are shown in Table 3.35. It can be seen that the real time 1 and 14 day forecasts are comparable to those obtained when model performance was assessed at various stages of learning (Tables 3.33 and 3.34). A plot of the real time 14 day forecast is shown in Figure 3.74.

Table 3.35: Real Time Forecasts Obtained Using Model SMB_U_A_15_06_91

Forecasting period (days)	RMSE (EC units)	AAPE (%)	AAE (EC units)
1	20.5	2.6	14.8
5	46.6	5.0	30.8
14	82.1	10.2	61.7

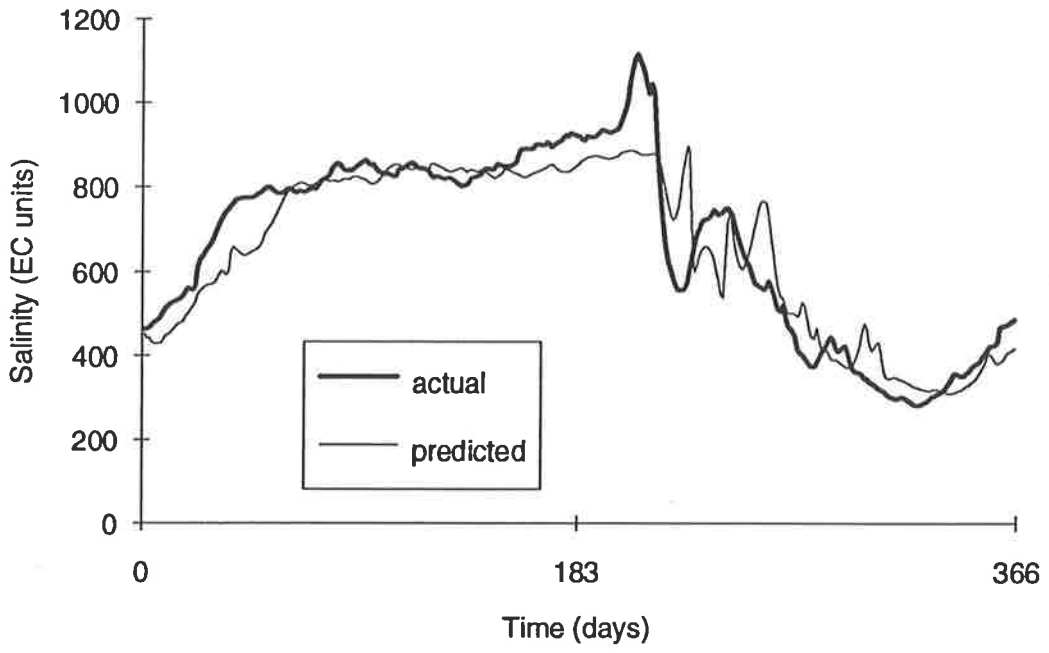


Figure 3.74: Real Time Forecast of Salinity in the River Murray at Murray Bridge Obtained Using Model SMB_U_A_15_06_91 - 1991 - 14 Days in Advance

3.5 Development of Multivariate Neural Network Model

3.5.1 Introduction

In this section, the development of the multivariate ANN (MANN) model for the prediction of salinity in the River Murray at Murray Bridge is described. The objectives are:

1. To develop the best possible model for forecasting salinity in the River Murray at Murray Bridge 14 days in advance.
2. To obtain 1, 5 and 14 day forecasts of salinity at Murray Bridge, which can be compared with those obtained using UANN, univariate and multivariate time series models.
3. To address the following issues in relation to ANN models:
   - How does the training data used affect network performance?
   - Can ANNs be used to extract information about the relationship between input and output variables?
   - What is the effect of adding less significant inputs on network performance?
   - What is the effect of having multiple network outputs?
   - What is the effect of internal parameters and geometry on network performance?
   - How does a network behave during training, especially as it approaches a local minimum in the error surface?

The steps shown in Figure 2.18 were followed to develop the various MANN models. The raw data were used in conjunction with hidden layer nodes and the direct forecasting technique, as they gave better results, and were simpler to use, when developing the UANN model (Section 3.4). A discussion of the plots of the input time series is given in Section 3.3.3.

3.5.2 Determination of Model Inputs

The procedure used for determining an appropriate set of model inputs is shown in Figure 3.75.

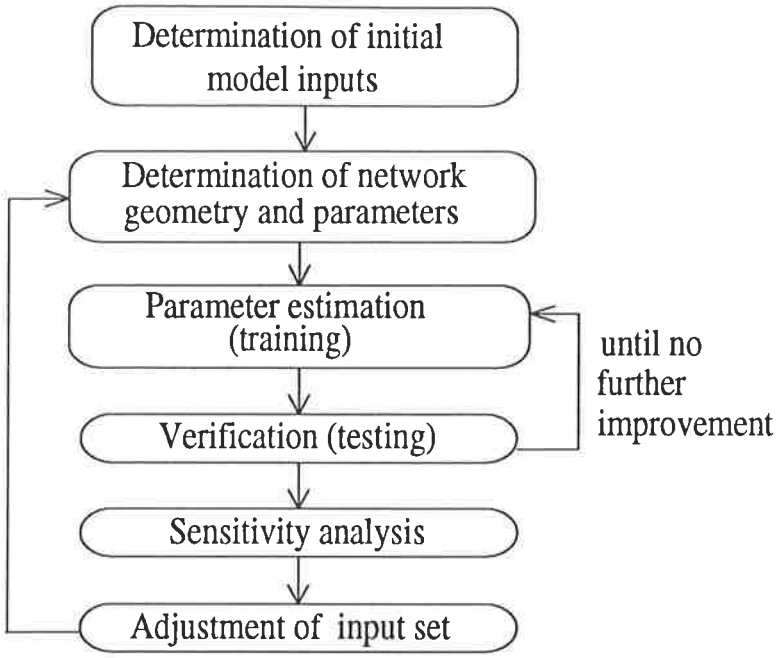


Figure 3.75: Procedure for Determining Appropriate Model Inputs

3.5.2.1 Determination of Initial Model Inputs

The initial model inputs were obtained by examining the underlying physical processes causing high salinities at Murray Bridge. As discussed in Section 3.3.3, the available data effectively allow the modelling of salinity transport as well as groundwater accessions.

Input data to account for salinity transport

Salinity transport is dependent on flow and upstream salinities (Jakeman et al., 1986). The time taken for salt to travel from one location to another is a function of flow. Salinity travel times at various flows between a number of locations in the South Australian reaches of the Murray were provided by the EWS (Table 3.36). In order to obtain 14 day forecasts of salinity at Murray Bridge, the salinity travel times between a particular upstream location and Murray Bridge have to be greater than 14 days. In accordance with Table 3.36, the salinity travel time from Mannum to Murray Bridge is 40 days at a flow of 4000 ML/day and 14 days at a flow of 6500 ML/day. Consequently, in order to cater for the range of flows from 4000 ML/day to 6500 ML/day, salinities from Mannum at lags 1, 2, 3, ..., 27 have to be included in the input set to obtain 14 day forecasts of salinity at Murray Bridge (i.e. at time t+13). Once the flow exceeds 6500 ML/day, the salinity travel times from Mannum to Murray Bridge are less than 14 days, so that salinities from a site upstream of Mannum are required to

produce a 14 day forecast at Murray Bridge. The lags of the salinities required from various upstream locations to produce 14 day forecasts at Murray Bridge at various flows are summarised in Table 3.37. The corresponding lags of the initial model inputs chosen are also shown in Table 3.37. It should be noted that the lags used as model inputs were slightly greater than those suggested by the salinity travel times to ensure that all significant inputs were included. It should also be noted that salinity values at Mannum were used at two-daily intervals in an attempt to reduce the total number of inputs. This is reasonable, as salinity travel times from Mannum to Murray Bridge are greater than 14 days at periods of low flow, during which changes in salinity occur slowly. In addition to the inputs shown in Table 3.37, salinities at Murray Bridge at lags 1, 2, 3, ..., 17 were included to give an indication of the short term trend of salinity at Murray Bridge. The choice of a maximum lag of 17 was arbitrary.

Flows at Lock 1 were chosen to model salinity transport, as this is the flow record closest to Murray Bridge. Close inspection of the plot of flow at Lock 1 Lower versus salinity at Murray Bridge (Figure 3.42) indicates that the major variations in flow lead the major variations in salinity by approximately 18 days. Consequently, it was decided to include flows at lags -23, -22, -21,..., -12, -11, which are centred around the lead time of 18 days (i.e. lag -17). The total number of flow inputs was chosen to cover a reasonable range of flows. Future values of flow can be obtained using an existing flow forecasting model (Water Studies Pty. Ltd., 1992). However, in this study, the actual values of future flows were used.

Table 3.36: Approximate Salinity Travel Times

Flow (ML/day)	Travel time (days)				
	Loxton to Waikerie	Waikerie to Morgan	Morgan to Mannum	Mannum to Murray Bridge	Total
4000	N/A	N/A	N/A	40.0	40.0
6500	N/A	N/A	N/A	14.0	14.0
7000	N/A	N/A	26.0	12.5	38.5
16000	N/A	N/A	10.0	4.0	14.0
17000	N/A	2.5	9.5	4.0	16.0
21000	N/A	2.5	8.0	3.5	14.0
22000	4.5	2.5	8.0	3.0	18.0
30000+	3.0	2.0	7.0	2.5	14.5

Table 3.37: Lags of Salinity Inputs Chosen to Cater for Salinity Transport

Site	Lags of salinity required for 14 day forecasts at Murray Bridge	Lags of salinity chosen as model inputs
Mannum	1, 2, ..., 27	1, 3, 5, ..., 33
Morgan	1, 2, ..., 26	1, 2, 3, ..., 31
Waikerie	1, 2, 3	1, 2, 3, ..., 11
Loxton	1, 2, ..., 5	1, 2, 3, ..., 11

Input data to account for groundwater accessions

As discussed in Section 3.3.3, groundwater accessions are a function of flow and river level. No information was available to suggest which lags of flows and river levels are critical. It was therefore decided to include level data at all available locations (with the exception of Overland Corner) at lags 1, 6, and 11, in order to cover a time span that is reasonable without having too many model inputs. Future values of level data (lags -19, -14 and -9) were included at Overland Corner to see whether future values of level are significant. It should be noted that future values of level could only be obtained at Overland Corner and Lock 1 Lower, as rating curves exist at these locations, relating flow to river level, so that flow forecasts can be used to obtain level forecasts. As with the level data, the flow inputs were chosen to cover a reasonable time span. Flow data at Lock 1 at lags -9, -4, ..., 36, 41 and at Overland Corner at lags -9, -4, ..., 16, 21 were included in the initial training / testing set.

The initial model inputs chosen (training / testing set 07), as discussed above, are summarised in Table 3.38. The total number of inputs chosen was 141. The only network output chosen was that of salinity at Murray Bridge at lag -13.

Table 3.38: Training / Testing Sets Used

	Training / testing set 07		Training / testing set 08		Training / testing set 09		Training / testing set 10	
Variable / Location	Lags (days)	Total No.	Lags (days)	Total No.	Lags (days)	Total No.	Lags (days)	Total No.
SMB	1, 2, ..., 16	16	1, 3, ..., 9	5	1, 2, 3, 4	3	1, 3, ..., 11	6
SMN	1, 3, ..., 33	17	1, 3, ..., 15	8	1, 3, ..., 15	8	1, 3, ..., 15	8
SMO	1, 2, ..., 31	31	1, 2, 3	3	1, 2, 3	3	1, 3, ..., 15	8
SWE	1, 2, ..., 11	11	1, 2, ..., 7	7	1, 2, 3	3	1, 2, ..., 5	5
SLO	1, 2, ..., 11	11	1, 2, ..., 6	6	1, 2, ..., 5	5	1, 2, ..., 5	5
FL1L	-23, ..., -11	13	-23, ..., -20	4				
	-9, -4, .., 41	11	-9, -4, .., -6	4				
			41	1				
FOC	-9, -4, .., 21	7	-9, -4, .., 21	7	-15, -13, .., -1	8	-19, -17, .., 7	14
LMB	1, 6, 11	3	1, 6, 11	3	1, 3, ..., 15	8		
LMN	1, 6, 11	3	1, 6, 11	3				
LL1U	1, 6, 11	3	1, 6, 11	3	6, 11, 16	3		
LL1L	1, 6, 11	3	1, 6, 11	3	6, 11, 16	3	-3, -1, ..., 5	5
LMO	1, 6, 11	3	1, 6, 11	3				
LWE	1, 6, 11	3	1, 6, 11	3				
LOC	-19, -14, -9	3	-19, -14, -9	3	-14, -12, .., -4	6		
LLO	1, 6, 11	3	1, 6, 11	3				
Total number	141		69		51		51	

3.5.2.2 Choice of Internal Parameters and Network Geometry

Choosing a successful network geometry is highly problem dependent. As discussed in Section 2.1.8.3, Maren et al. (1990) suggest using 2 hidden layers when the outputs need to be continuous functions of the inputs. Consequently, it was decided to use 2 hidden layers. Guidance as to how many hidden layer nodes should be used is given by several authors, as discussed in Section 2.1.8.3. A number of these guidelines (e.g. Baum and Haussler, 1989; Weigend et al., 1990) relate network size to the number of training samples available. However, as pointed out by Chakraborty (1992), such methods are too restrictive when a small number of training samples are available, as is the case here. Hecht-Nielsen (1987) suggests an upper limit of $2N^I+1$ hidden layer nodes, where N^I is the number of inputs. However, in many practical applications, the optimum number of hidden layer nodes was found to be considerably less (e.g. DeSilets

et al., 1992). Maren et al. (1990) indicate that for many applications, the optimum number of hidden layer nodes has been found to be less than the number of inputs. Taking the above information into account, it was decided to use 90 nodes in the first hidden layer. The ratio of first to second hidden layer nodes used was 3:1 in accordance with Kudrycki (1988). Consequently, 30 nodes were used in the second hidden layer, resulting in a 141-90-30-1 network (i.e. 141 inputs, 90 nodes in the first hidden layer, 30 nodes in the second hidden layer and 1 output).

The internal network parameters include the learning rate (η), momentum (μ), epoch size (ε), transfer function, error function, learning rule and initial weight distribution. The default values in the software package, which were determined using the experience gained from developing back-propagation models for a variety of applications (NeuralWare, Inc., 1991), were used to train all networks. The default values of the learning rate and momentum are shown in Table 3.39. The hyperbolic tangent transfer function, the normalised cumulative delta learning rule, the quadratic error function, as well as an epoch size of 16, were used. The initial weights were randomly distributed between +0.1 to -0.1.

It should be noted that different geometries and internal parameters were not used, as the main objective of this stage of the model development process is to determine an appropriate set of model inputs.

Table 3.39: Default Learning Rates (η) and Momentum Values (μ)

Layer	Parameter	Learn count				
		0 to 10,000	10,001 to 30,000	30,001 to 70,000	70,001 to 310,000	310,001 +
Output layer	η	0.15	0.075	0.01875	0.00117	0.00001
	μ	0.40	0.200	0.05000	0.00313	0.00001
Hidden layer 2	η	0.25	0.125	0.03125	0.00195	0.00001
	μ	0.40	0.200	0.05000	0.00313	0.00001
Hidden layer 1	η	0.30	0.150	0.03750	0.00234	0.00001
	μ	0.40	0.200	0.05000	0.00313	0.00001

3.5.2.3 Training / Testing

In order to make maximum use of the limited data set available, and to investigate the effect of using different years of data for training and testing on the generalisation ability

of the network, four networks were trained using training / testing set 07 (Table 3.38); each using different years of data for training and testing. In model SMB_M_A_16_07_88, the data from 1988 were used for testing while the remaining data were used in the training phase. Similarly, data from 1989 were used for testing in model SMB_M_A_16_07_89, data from 1990 were used for testing in model SMB_M_A_16_07_90 and data from 1991 were used for testing in model SMB_M_A_16_07_91. In each case, the remaining data were used for training.

The testing interval used was 10,000. When training the four models, less than 50,000 training samples had to be presented to the networks before a plateau was reached in the RMSE of the test set. The RMSEs at the various stages of learning are shown in Appendix E. The time taken for each network to process 10,000 training samples was 27 minutes.

The RMSEs, AAPEs and AAEs of the best 14 day forecasts obtained for models SMB_M_A_16_07_88, SMB_M_A_16_07_89, SMB_M_A_16_07_90 and SMB_M_A_16_07_91 are shown in Table 3.40. It can be seen that the performance of the models is generally good, with a RMSE of 53.3 EC units, an AAPE of 7.5%, and an AAPE of 38.3 EC units, averaged over the 4 years. However, model performance is significantly influenced by the data used for training and the data used for testing. The best forecast was obtained for 1988 with a RMSE of 47.7 EC units, while the worst forecast was obtained for 1989 with a RMSE of 57.8 EC units.

Table 3.40: Best 14 Day Forecasts Obtained for the Various Training / Testing Sets

Training / testing set	Performance measure	Year predicted				Average
		1988	1989	1990	1991	
07	RMSE (EC units)	47.7	57.8	50.0	57.7	53.3
	AAPE (%)	5.3	8.0	8.6	8.2	7.5
	AAE (EC units)	33.0	37.9	37.6	44.5	38.3
08	RMSE (EC units)	44.4	49.3	44.5	52.0	47.6
	AAPE (%)	5.1	6.4	7.4	7.0	6.5
	AAE (EC units)	31.4	31.3	32.3	37.7	33.2
09	RMSE (EC units)	42.6	55.3	45.2	44.4	46.9
	AAPE (%)	4.8	6.5	7.5	6.6	6.4
	AAE (EC units)	31.2	32.3	32.6	33.8	32.5
10	RMSE (EC units)	44.8	49.3	44.5	45.8	46.1
	AAPE (%)	5.3	6.8	7.0	6.5	6.4
	AAE (EC units)	31.0	32.4	32.2	34.0	32.4

3.5.2.4 Sensitivity Analysis

Although ANNs belong to the class of data driven approaches, it is important to keep the number of inputs, and hence the size of the network, to a minimum, as larger networks take longer to train, require more data to efficiently estimate the connection weights and have an increased number of local minima in the error surface. A sensitivity analysis was carried out using models SMB_M_A_16_07_88, SMB_M_A_16_07_89, SMB_M_A_16_07_90 and SMB_M_A_16_07_91 to determine the relative significance of each of the model inputs, with the aim of deleting those inputs that do not have a significant effect on model performance. In order to determine which inputs to retain and which inputs to delete, plots of the sensitivities of the inputs were inspected. A typical plot of the relative significance of the salinity inputs at Mannum is shown in Figure 3.76. No fixed level was used to distinguish between significant and non-significant inputs. The sensitivities were used as a guide to decide which inputs to retain and which to delete by applying some degree of judgement. In the case of salinity at Mannum, the inputs at lags 1, 3, ..., 15 were deemed to be significant and were retained for training / testing set 08 (Table 3.38).

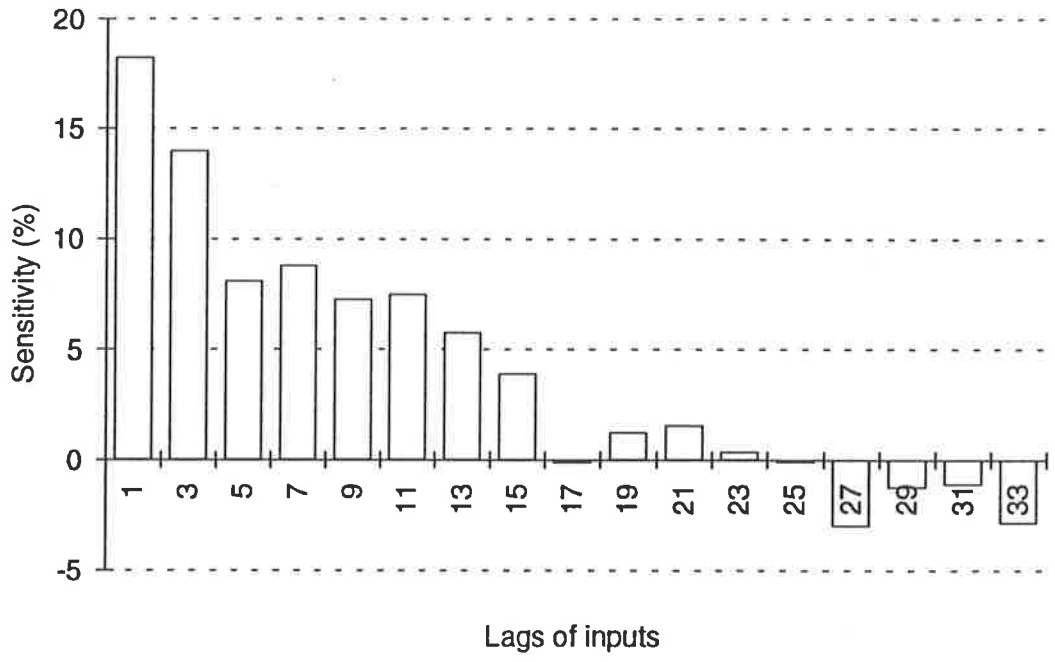


Figure 3.76: Typical Sensitivities of Salinity Inputs from Mannum

The plots of the sensitivities for the remaining salinity inputs and the flow inputs (not shown here) were also inspected, and appropriate changes made to training / testing set 07 (Table 3.38). It should be noted that the frequency of the inputs of salinity at Murray

Bridge was reduced to alternate days in an attempt to reduce the number of inputs. No changes were made to the level inputs, so that the effects of changing the salinity and flow inputs could be assessed. The total number of inputs was reduced from 141 to 69.

The effect of using training / testing set 08 was assessed in models SMB_M_A_17_08_88, SMB_M_A_17_08_89, SMB_M_A_17_08_90 and SMB_M_A_17_08_91. Using the guidelines described above, the network geometry was chosen to be 69-45-15-1. The testing interval used was 5,000. When training the four models, a plateau in the RMSE of the test set was reached after the presentation of approximately 50,000 training samples. It should be noted that the connection weights were re-initialised before training commenced in order to get an indication of the total time taken for training networks of different sizes. The RMSEs at the various stages of learning are shown in Appendix E. By reducing the network size from 141-90-30-1 to 69-45-15-1, the time taken to process 10,000 training samples was reduced from 27 minutes to 7 minutes. In addition, the RMSE, averaged over the four models, was reduced from 53.3 EC units to 47.6 EC units, the AAPE, averaged over the 4 models, was reduced from 7.5% to 6.5% and the AAE, averaged over the 4 models, was reduced from 38.3 EC units to 33.2 EC units (Table 3.40).

A sensitivity analysis, similar to that described above, was carried out using models SMB_M_A_17_08_88, SMB_M_A_17_08_89, SMB_M_A_17_08_90 and SMB_M_A_17_08_91. The resulting changes made to training / testing set 08 are shown in Table 3.38 (training / testing set 09). It should be noted that all flows at Lock 1 were removed, as they are very similar to flows at Overland Corner. Additional level data were included at Murray Bridge and Overland Corner, and the frequency of these inputs was reduced from 5 days to 2 days. The total number of inputs was reduced from 69 to 51.

The effect of using training / testing set 09 was assessed in models SMB_M_A_18_09_88, SMB_M_A_18_09_89, SMB_M_A_18_09_90 and SMB_M_A_18_09_91. Using the guidelines described above, the network geometry was chosen to be 51-45-15-1. The testing interval used was 5,000. When training the four models, a plateau in the RMSE of the test set was reached after the presentation of approximately 50,000 training samples. The RMSEs at the various stages of learning are shown in Appendix E. By reducing the network size from 69-45-15-1 to 51-45-15-1, the time taken to process 10,000 training samples was reduced from 7 minutes to 6 minutes. In addition, the RMSE, averaged over the four models, was reduced from 47.6 EC units to 46.9 EC units, the AAPE, averaged over the 4 models, was reduced from

6.5% to 6.4% and the AAE, averaged over the 4 models, was reduced from 33.2 EC units to 32.5 EC units (Table 3.40).

Inspections of the data plots of the input time series (shown in Section 3.3.3.1) were used in conjunction with sensitivity analyses carried out on models SMB_M_A_18_09_88, SMB_M_A_18_09_89, SMB_M_A_18_09_90 and SMB_M_A_18_09_91 to adjust training / testing set 09. The number of inputs of salinity at Murray Bridge, Morgan and Waikerie, as well as flow at Overland Corner were increased. All level data, with the exception of levels at Lock 1 Lower, were removed, as there appears to be a strong correlation between the level data at all locations. The changes made to training / testing set 09 are shown in Table 3.38 (training / testing set 10). The total number of inputs used remained at 51.

The effect of using training / testing set 10 was assessed in models SMB_M_A_18_10_88, SMB_M_A_18_10_89, SMB_M_A_18_10_90 and SMB_M_A_18_10_91. The internal network parameters and geometry were identical to those used for models SMB_M_A_18_09_88, SMB_M_A_18_09_89, SMB_M_A_18_09_90 and SMB_M_A_18_09_91. The testing interval used was 5,000. When training the four models, a plateau in the RMSE of the test set was reached after the presentation of approximately 50,000 training samples. The RMSEs at the various stages of learning are shown in Appendix E. By using training / testing set 10 instead of training / testing set 09, the RMSE, averaged over the four models, was reduced from 46.9 EC units to 46.1 EC units, the AAPE, averaged over the 4 models, remained constant at 6.4 % and the AAE, averaged over the 4 models, was reduced from 32.5 EC units to 32.4 EC units (Table 3.40). The best forecast was obtained for 1990, with an RMSE of 44.5 EC units , and the worst forecast was obtained for 1989, with an RMSE of 49.3 EC units. It is interesting to note that the difference in the RMSEs between the best and the worst forecasts obtained using training / testing set 10 (i.e. 4.8 EC units) is considerably less than that obtained using training / testing set 07 (i.e. 10.1 EC units), indicating that the generalisation ability of the models trained with training / testing set 10 is considerably better. One of the reasons for this could be associated with the difficulty of finding near-optimum local minima in the error surface for larger networks.

It should be noted that there is some disagreement between the three performance measures (i.e. RMSE, AAPE and AAE) as to which of the models performs the best for any particular year of testing (Table 3.40). In addition, when considering the RMSEs of the various models, the best forecasts for the different years are obtained using different input sets (Table 3.40). However, when the RMSEs, AAPEs and AAEs are averaged

over the 4 years used for forecasting (i.e. 1988-1991), the three performance measures are in agreement, and there is an increase in performance from training / testing set 07 to training / testing set 10. This highlights the fact that the data used for training and for testing can have a significant effect on the results obtained, and stresses the importance of obtaining independent forecasts for a number of years, so that the generalisation ability of the models can be accurately assessed.

It was decided to adopt training / testing set 10, as it produced the best 14 day forecasts of the training / testing sets considered and, judging from the consistency of the results obtained using training / testing sets 08, 09 and 10, it seems unlikely that further changes in the training / testing set would result in significant improvements in generalisation ability or training speed.

By inspecting training / testing set 10, it can be seen that the lags of the salinity inputs that were found to be significant (Table 3.38) are in good agreement with the inputs suggested by the salinity travel times measured by the EWS (Table 3.37). Fewer model inputs were found to be significant at Mannum and Morgan than the salinity travel times would suggest. However, salinities from these locations are used to obtain forecasts at Murray Bridge at times of low flow, when changes in salinity are small. Consequently, if the salinities at greater lags are not significantly different from those at smaller lags, they do not provide any new information and are not required by the model. The salinity inputs that were found to be significant at Waikerie and Loxton are in excellent agreement with the salinity travel times.

The performance of the models trained with the different input sets indicates that flow and level inputs are only required from one location, although saline groundwater inflows occur all along the river. This suggests that the flows and levels at one location are indicative of those at all other locations, and that the relationship between river level, flow and saline accessions is similar along the length of the river.

3.5.2.5 Effect of the Number of Outputs

The effect of including multiple network outputs was assessed in models SMB_M_A_19_11_88, SMB_M_A_19_11_89, SMB_M_A_19_11_90 and SMB_M_A_19_11_91, which are identical to models SMB_M_A_18_10_88, SMB_M_A_18_10_89, SMB_M_A_18_10_90 and SMB_M_A_18_10_91, with the exception of the inclusion of additional outputs at lags -1, -3, -5, -7, -9 and -11.

The testing interval used was 5,000. When training the four models, less than 50,000 training samples had to be presented to the networks before a plateau was reached in the RMSE of the test set. The RMSEs at the various stages of learning are shown in Appendix E. The time taken for the networks to process 10,000 training samples remained at 6 minutes.

The RMSEs, AAPEs and AAEs of the best 14 day forecasts obtained are shown in Table 3.41. It can be seen that the inclusion of additional inputs did not have a significant impact on overall model performance.

Table 3.41: Results Obtained Using a Different Number of Outputs

Number of outputs	Performance measure	Year predicted				Average
		1988	1989	1990	1991	
7	RMSE (EC units)	41.0	48.4	46.4	48.4	46.1
	AAPE (%)	4.7	7.1	8.3	6.8	6.7
	AAE (EC units)	28.9	34.5	33.9	37.8	33.8
1	RMSE (EC units)	44.8	49.3	44.5	45.8	46.1
	AAPE (%)	5.3	6.8	7.0	6.5	6.4
	AAE (EC units)	31.0	32.4	32.2	34.0	32.4

3.5.2.6 Results / Discussion

Plots of the 14 day forecasts obtained using models SMB_M_A_18_10_88, SMB_M_A_18_10_89, SMB_M_A_18_10_90 and SMB_M_A_18_10_91 are shown in Figures 3.77 to 3.80. It can be seen that the forecasts obtained are very good, as they forecast all major variations in salinity without appreciable lag. Some of the small peaks in salinity were not predicted. However, these are not significant and might be due to large saline accessions other than groundwater inflows (e.g. saline inflows from irrigation drains), for which no data are available. In addition, most of these peaks occur at times of high flow, during which salinities at locations further upstream (e.g. Loxton) are required to produce 14 day forecasts at Murray Bridge. However, at sites further upstream, most variations in salinity are less pronounced or even absent, as discussed in Section 3.3.3.1, making it difficult to forecast them.

The models underestimate the maximum value of some of the major peaks in salinity. However, this is not crucial, as the absence or presence of a major peak provides sufficient information for the adjustment of the pumping schedules.

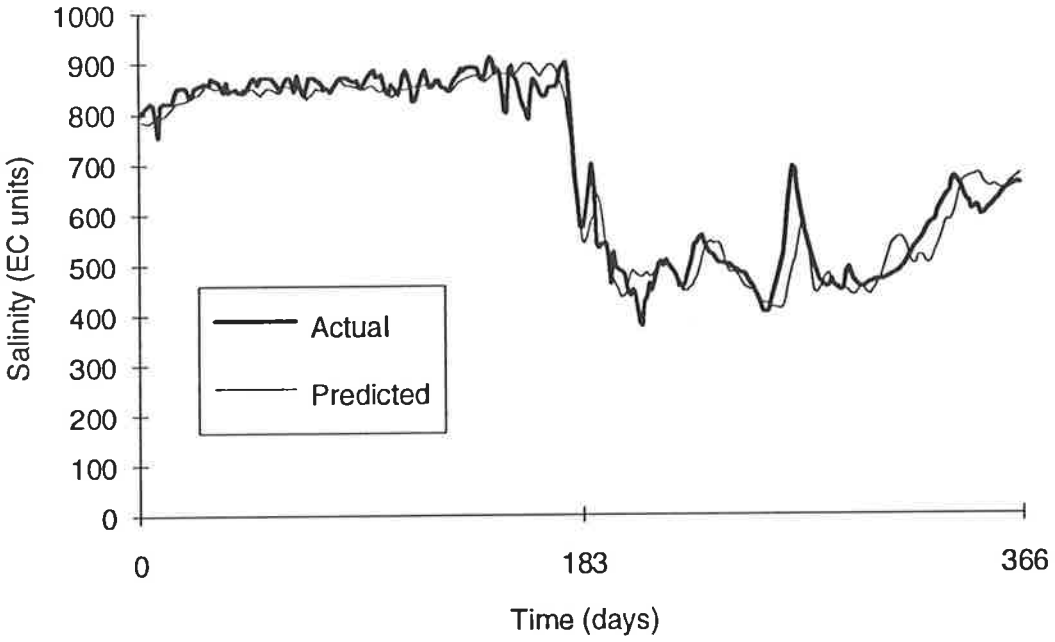


Figure 3.77: Best 14 Day Forecast of Salinity at Murray Bridge for 1988 - Model SMB_M_A_18_10_88

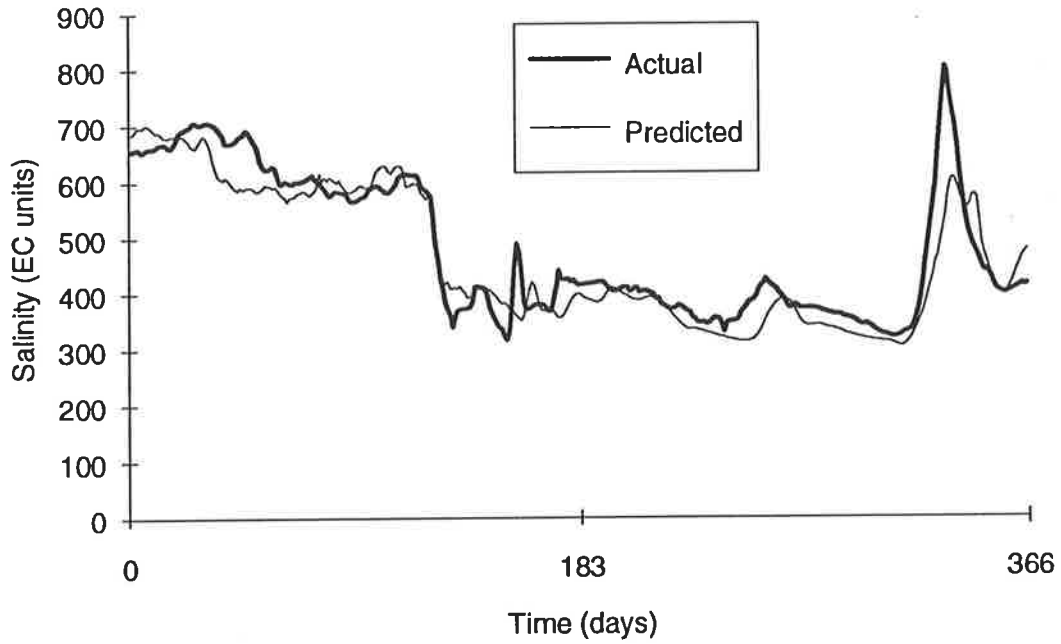


Figure 3.78: Best 14 Day Forecast of Salinity at Murray Bridge for 1989 - Model SMB_M_A_18_10_89

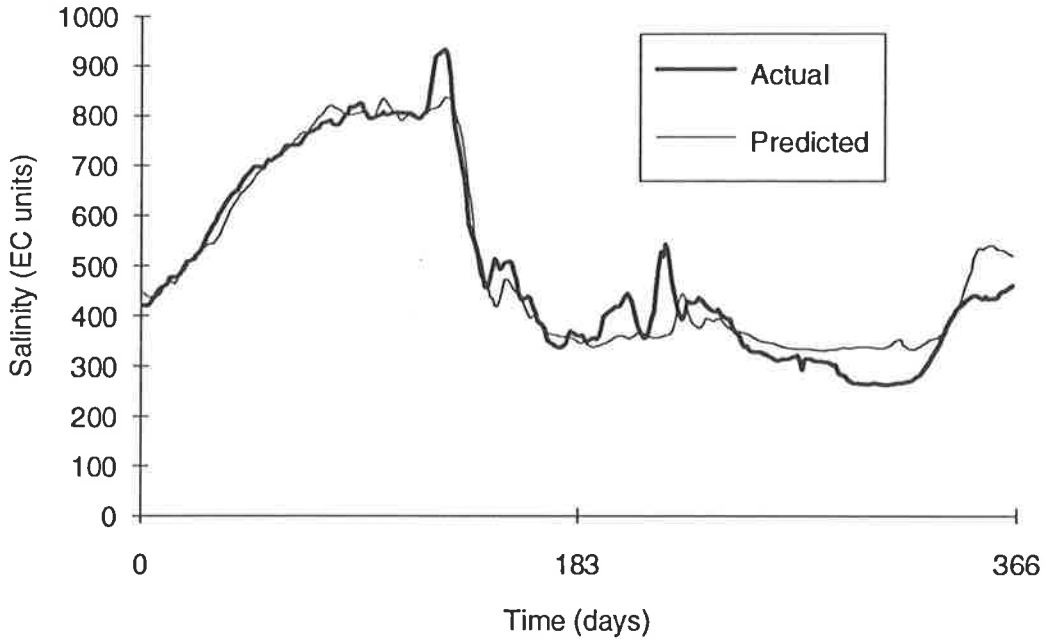


Figure 3.79: Best 14 Day Forecast of Salinity at Murray Bridge for 1990 - Model SMB_M_A_18_10_90

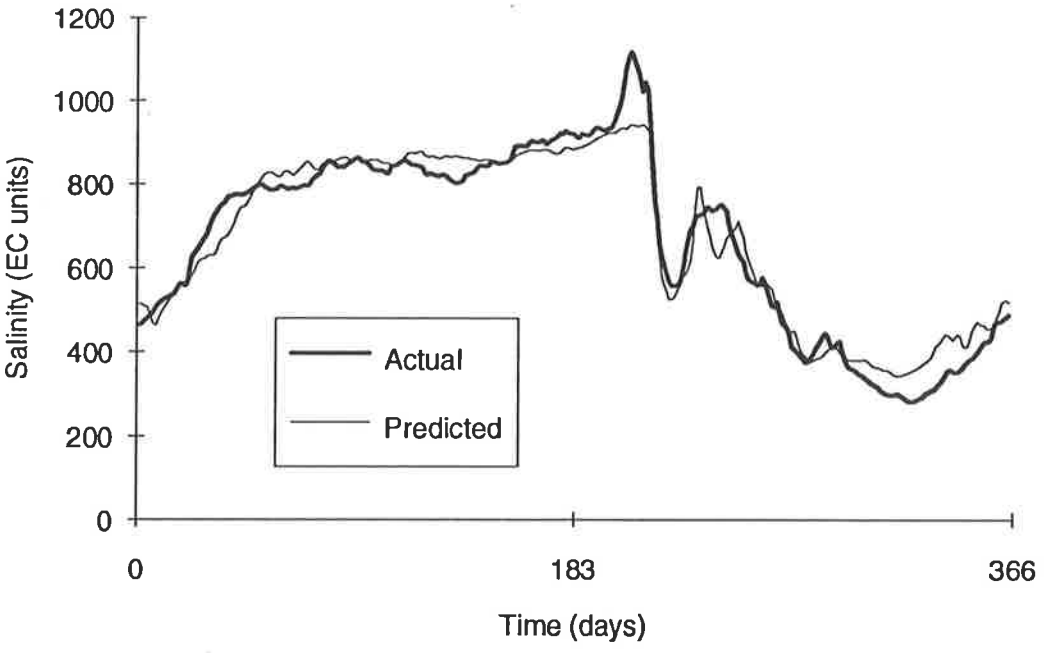


Figure 3.80: Best 14 Day Forecast of Salinity at Murray Bridge for 1991 - Model SMB_M_A_18_10_91

The predictive ability of the models is generally worse in the latter portion of each year. This may be attributed to the fact that these time periods follow periods of high

flow (Figures 3.42 and 3.43). As discussed in Section 3.2.1.1, the mechanisms by which salt enters the river are different after the passing of the flood hydrograph, and the available input data does not cater for some of these.

3.5.3 Investigation of the Effects of Internal Parameters and Network Geometry

3.5.3.1 Introduction

One of the most difficult, and least understood, tasks in the design of back-propagation networks is the choice of adequate internal network parameters and an appropriate network geometry. Although some guidance is available for choosing appropriate network parameters and geometries (Section 2.1.8.3), they generally have to be determined using a trial and error approach.

Network parameters and network geometry can have a significant impact on the way networks learn, as indicated in Figure 3.81. The network outputs obtained, for a given set of inputs, depend on the values of the connection weights, the transfer function and network geometry. The predicted outputs are then compared with the actual (desired) outputs using an error function. The global error function ($E^G(t)$) most commonly used is the quadratic error function (Equation 2.15).

The connection weights are then adjusted using a form of the generalised delta learning rule in an attempt to reduce the error function. The amount by which each connection weight is adjusted depends on the learning rate (η), the momentum value (μ), the epoch size (ε), the derivative of the transfer function and the node output.

In this section, the effects of the internal parameters and network geometry on learning speed and generalisation ability are investigated for the case study under consideration. The objectives of this study include:

1. To gain a better understanding of the impact of the internal parameters and network geometry on learning speed and generalisation ability.
2. To obtain the "optimum" internal parameters and network geometry for the case study under consideration.
3. To investigate the generalisation ability of back-propagation networks at various stages of learning, especially as they approach a local minimum in the error surface, under a wide range of operating conditions.
4. To try and develop some guidelines for optimising network performance, which can be used for other case studies.

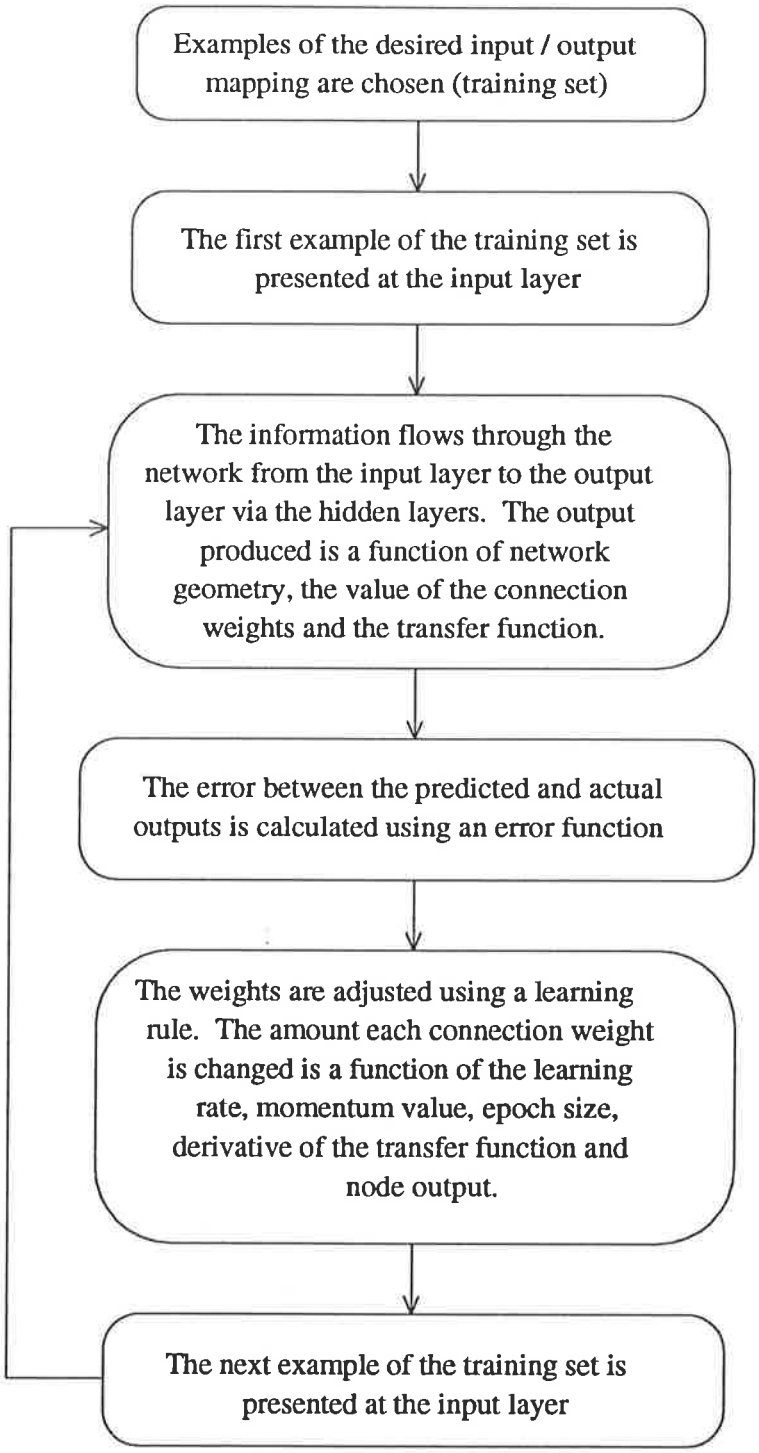


Figure 3.81: Summary of the Back-Propagation Training Process

3.5.3.2 Method

In each set of tests, one parameter was varied and all other parameters were kept constant in order to assess the effect of changes in the parameter investigated only. Cross-validation was used to assess the generalisation ability of the networks at various stages of learning. Data from 1987, 1989, 1990 and 1991 were used for training, while

data from 1988 were used to test the performance of the models at different stages of learning. The 1988 data were used for testing, as all salinity, flow and level values in that year (the test set) lie between the values of the other years (the training set), and good forecasts were obtained for 1988 in Section 3.5.2. It should be noted that all plots of RMSEs at various stages of learning shown in this section were obtained using the test set.

An epoch size (ε) of 5, a momentum value (μ) of 0.6, a learning rate (η) of 0.005, the hyperbolic tangent transfer function, the normalised cumulative delta learning rule (NCDE) and the standard (quadratic) error function were used unless stated otherwise. The number of nodes used in the first hidden layer was 45 and the number of nodes used in the second hidden layer was 15 unless stated otherwise. Training / testing set 10 was used to train all models.

3.5.3.3 Effect of Internal Parameters on Network Performance

In this section, the effects of the epoch size, momentum, learning rate, transfer function, error function and initial weight distribution on training speed and generalisation ability were investigated.

Epoch size

As discussed in Section 2.1.8.5, the epoch size (ε) is defined as the number of training samples presented to networks between weight updates. Using larger epoch sizes may speed up learning, as each weight update works more towards reaching the global minimum in the error surface (NeuralWare, Inc., 1991). However, this advantage may be lost if the epoch size is too large, as many more calculations need to be carried out for a single update. The normalised cumulative delta rule (NeuralWare, Inc., 1991) was used for this series of tests.

Epoch sizes of 5, 10, 16, 35, 85, 180, 365 and 1671 were used in this series of tests. It should be noted that an epoch size of 365 was used as there is a yearly cycle in the data, and an epoch size of 1671 was used as that is the number of samples in the training set. Initially, a learning rate (η) of 0.005 and a momentum (μ) of 0.6 were used. The models developed include model SMB_M_A_20_10_88 (ε = 5), model SMB_M_A_21_10_88 (ε = 10), model SMB_M_A_22_10_88 (ε = 16), model SMB_M_A_23_10_88 (ε = 35), model SMB_M_A_24_10_88 (ε = 85), model SMB_M_A_25_10_88 (ε = 180), model SMB_M_A_26_10_88 (ε = 365) and model

SMB_M_A_27_10_88 (ε = 1671). Details of the network parameters and geometries used for the various models are summarised in Appendix D.

The RMSEs at the various stages of learning are shown in Figures 3.82 and 3.83 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Table 3.42. It can be seen that for the training set used:

- There is no significant difference between the best forecasts obtained using different epoch sizes.
- Learning speed is much greater for the networks using smaller epoch sizes.
- There is a steady decrease in the RMSE for all networks until a local minimum "plateau" in the error surface is reached.

Table 3.42: Best Results for Models With Various Epoch Sizes (η = 0.005, μ = 0.6)

Model	Epoch size (ε)	RMSE (EC units)	Learn count
SMB_M_A_20_10_88	5	42.0	105,000
SMB_M_A_21_10_88	10	42.2	150,000
SMB_M_A_22_10_88	16	42.2	175,000
SMB_M_A_23_10_88	35	42.3	320,000
SMB_M_A_24_10_88	85	42.4	400,000
SMB_M_A_25_10_88	180	42.6	555,000
SMB_M_A_26_10_88	365	42.5	800,000
SMB_M_A_27_10_88	1671	42.3	1,871,520

Figure 3.84 indicates that fewer weight updates have to be performed to reach the local minimum when larger epoch sizes are used. However, as explained in Section 3.3.4.1, the error term used in the weight update equation by the normalised cumulative delta learning rule is equal to the individual errors produced by each training sample summed over one epoch, divided by the square root of the epoch size. Consequently, the size of the error term used is proportional to the square root of the epoch size, resulting in bigger changes in the connection weights per update for networks with bigger epoch sizes. In other words, for a given error, networks using larger epoch sizes may be considered to have larger "effective learning rates".

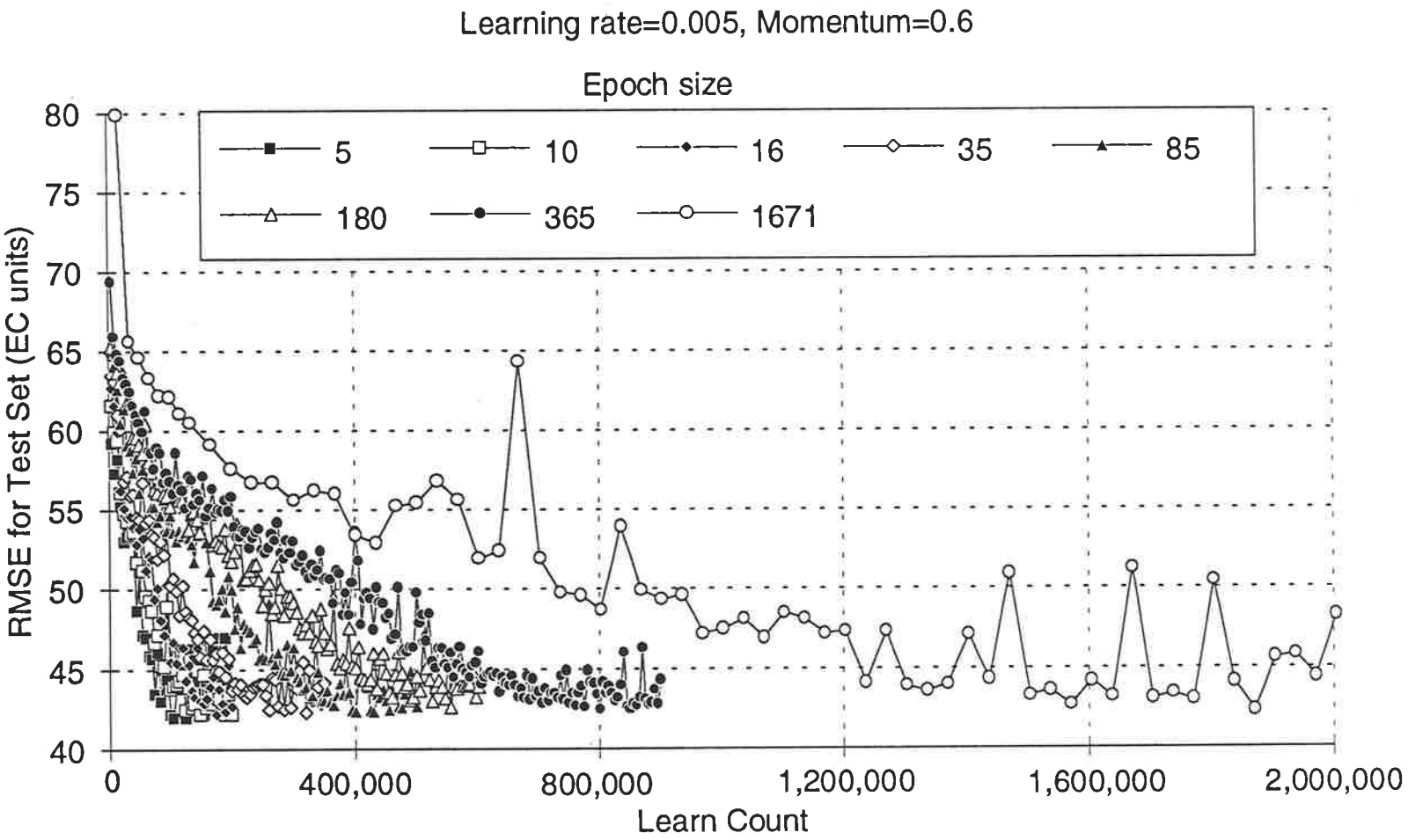


Figure 3.82: RMSE at Different Stages of Learning for Various Epoch Sizes

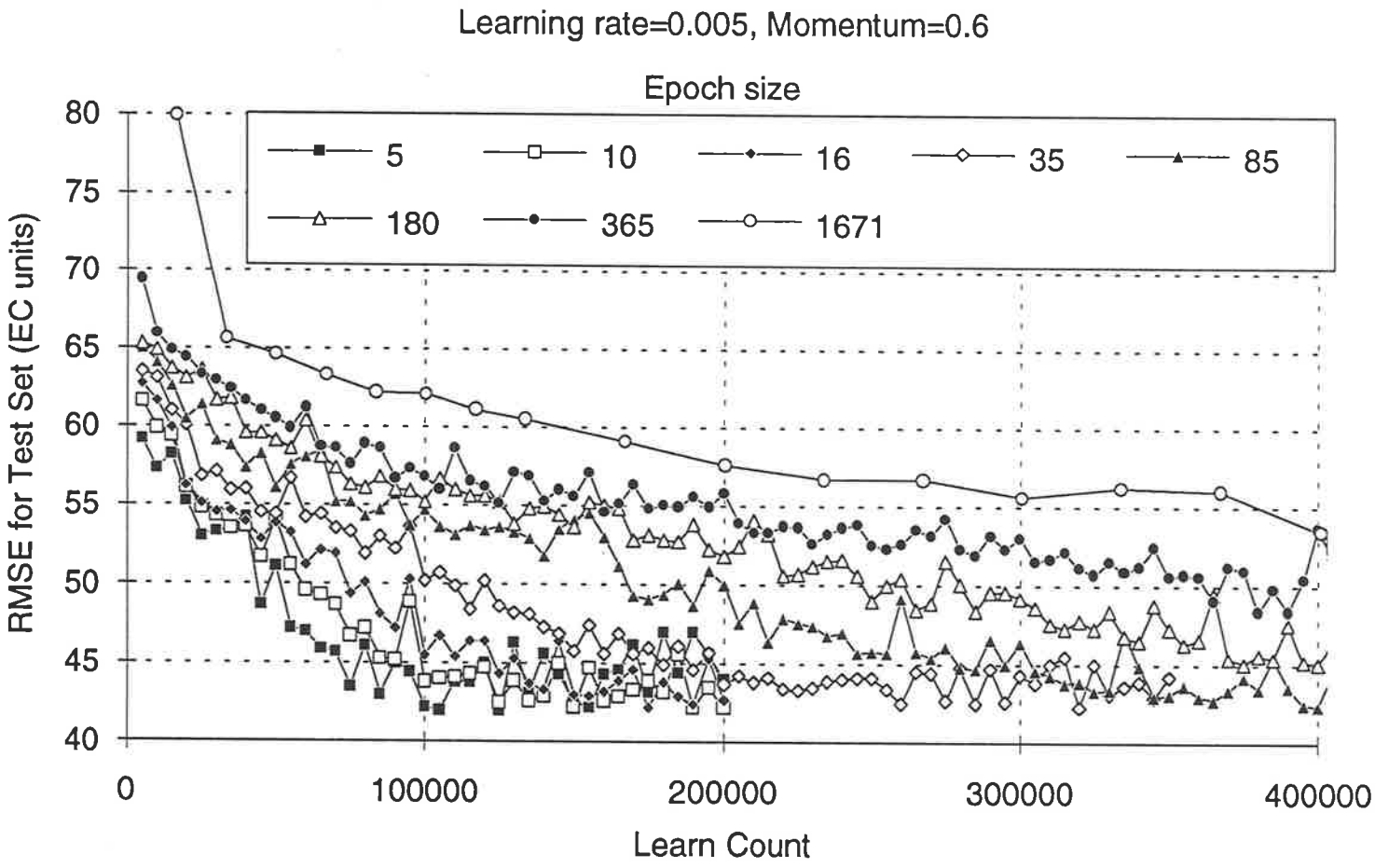


Figure 3.83: RMSE at Different Stages of Learning for Various Epoch Sizes (Up to a Learn Count of 400,000)

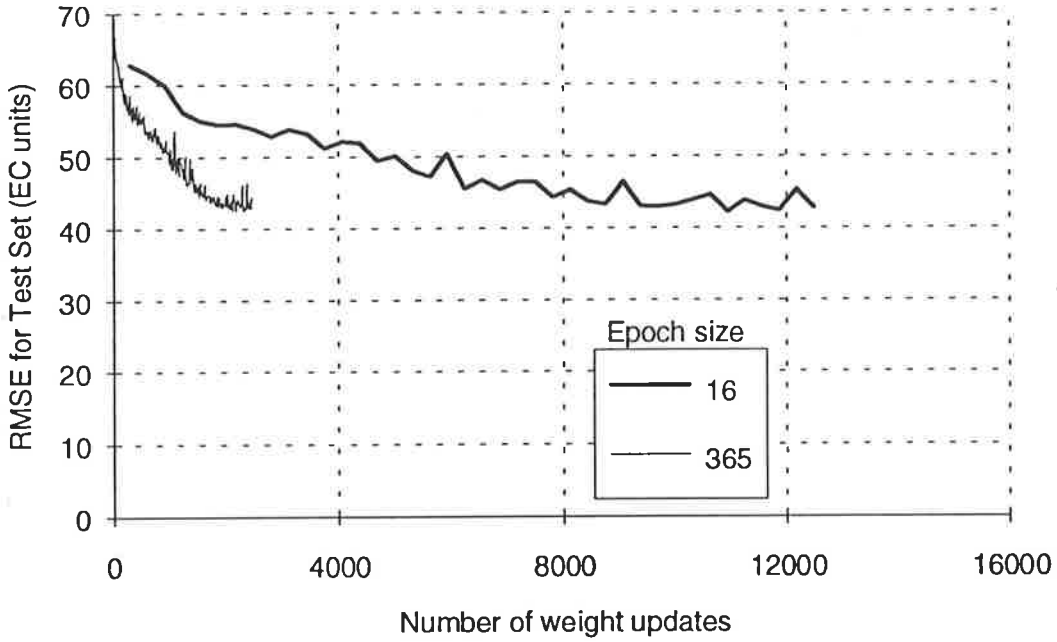


Figure 3.84: Change in RMSE versus Number of Weight Updates for Models Using Different Epoch Sizes

In order to obtain a fair comparison between the number of weight updates performed to reach the local minimum in the error surface for various epoch sizes, the number of weight updates performed have to be normalised by multiplying the number of weight updates by the square root of the epoch size. The plot of RMSE versus normalised number of weight updates for epoch sizes of 16 and 365 (Figure 3.85) shows that the normalised number of weight updates required to reach the local minimum in the error surface is the same regardless of epoch size. This indicates that for the training set considered, using a large epoch size does not contribute any more towards reaching the global minimum in the error surface than using a small epoch size. Consequently learning speed is much faster for networks with smaller epoch sizes, as the time taken to perform one weight update is considerably less.

The bigger steps in weight space taken by the networks with larger epoch sizes also account for the larger variations in the RMS prediction errors obtained for these networks (Figure 3.82).

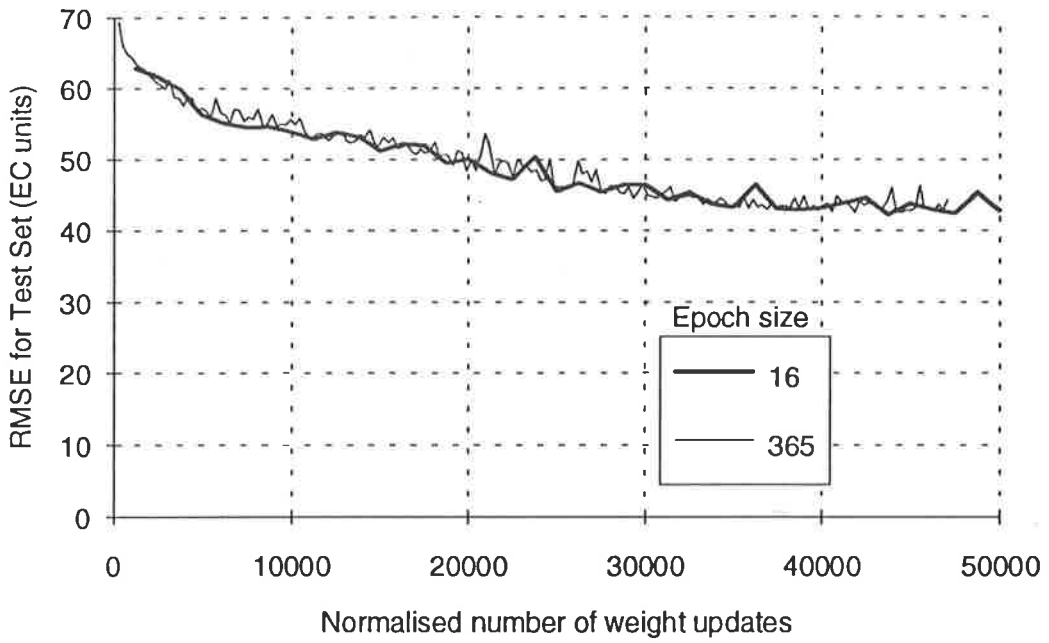


Figure 3.85: Change in RMSE versus Normalised Number of Weight Updates for Models SMB_M_A_22_10_88 and SMB_M_A_26_10_88

The above tests were repeated using a learning rate of 0.1 (instead of 0.005), in order to assess whether the effect of different epoch sizes is the same irrespective of the learning rate used. The models developed include model SMB_M_A_28_10_88 ($\varepsilon = 5$), model SMB_M_A_29_10_88 ($\varepsilon = 10$), model SMB_M_A_30_10_88 ($\varepsilon = 16$) and model SMB_M_A_31_10_88 ($\varepsilon = 35$). It should be noted that for networks with epoch sizes greater than 35, the transfer functions saturated and learning ceased as a result of the large steps taken in weight space. Details of the network parameters and geometries used for the various models are summarised in Appendix D.

The RMSEs at the various stages of learning are shown in Figure 3.86 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Table 3.43. As can be seen, the results are not as informative as those obtained with learning rates of 0.005 as:

- Networks could only be trained for a very narrow range of epoch sizes.
- The behaviour of the networks was less controlled as a result of the large steps taken in weight space, resulting in a larger variation in the best RMS prediction errors obtained for the various epoch sizes, as indicated in Table 3.43. In addition, continued training resulted in divergent behaviour (Figure 3.86).

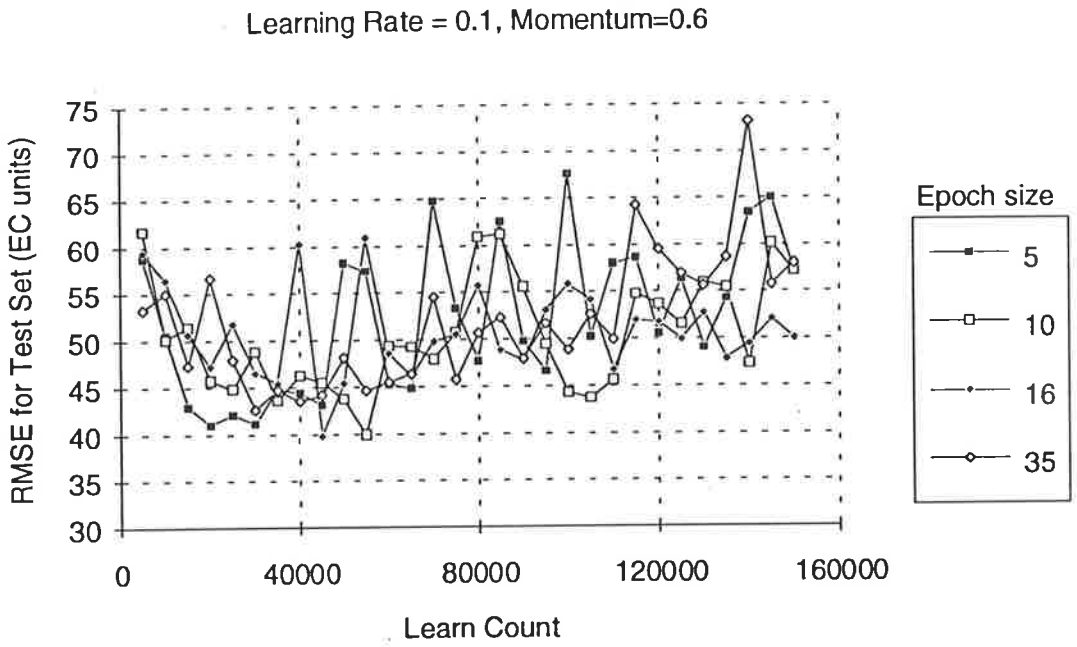


Figure 3.86: RMSE at Different Stages of Learning for Various Epoch Sizes ($\eta = 0.1$, $\mu = 0.6$)

Table 3.43: Best Results for Models With Various Epoch Sizes ($\eta = 0.1$, $\mu = 0.6$)

Model	Epoch size (ε)	RMSE (EC units)	Learn count
SMB_M_A_28_10_88	5	41.0	20,000
SMB_M_A_29_10_88	10	39.9	55,000
SMB_M_A_30_10_88	16	39.8	45,000
SMB_M_A_31_10_88	35	42.6	30,000

Models SMB_M_A_20_10_88 and SMB_M_A_26_10_88 were re-trained using a momentum value of 0.05 (rather than 0.6) in order to assess whether the effect of different epoch sizes was the same regardless of the momentum value used. An epoch size of 5 was used for model SMB_M_A_32_10_88 and an epoch size of 365 was used for model SMB_M_A_33_10_88 (Appendix D).

The RMSEs at the various stages of learning are shown in Figure 3.87 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Table 3.44. It can be seen that:

- There is no significant difference between the best forecasts obtained using epoch sizes of 5 and 365.
- Learning speed is much greater when using the smaller epoch size.

This confirms the results obtained using a momentum value of 0.6 (Models SMB_M_A_20_10_88 and SMB_M_A_26_10_88).

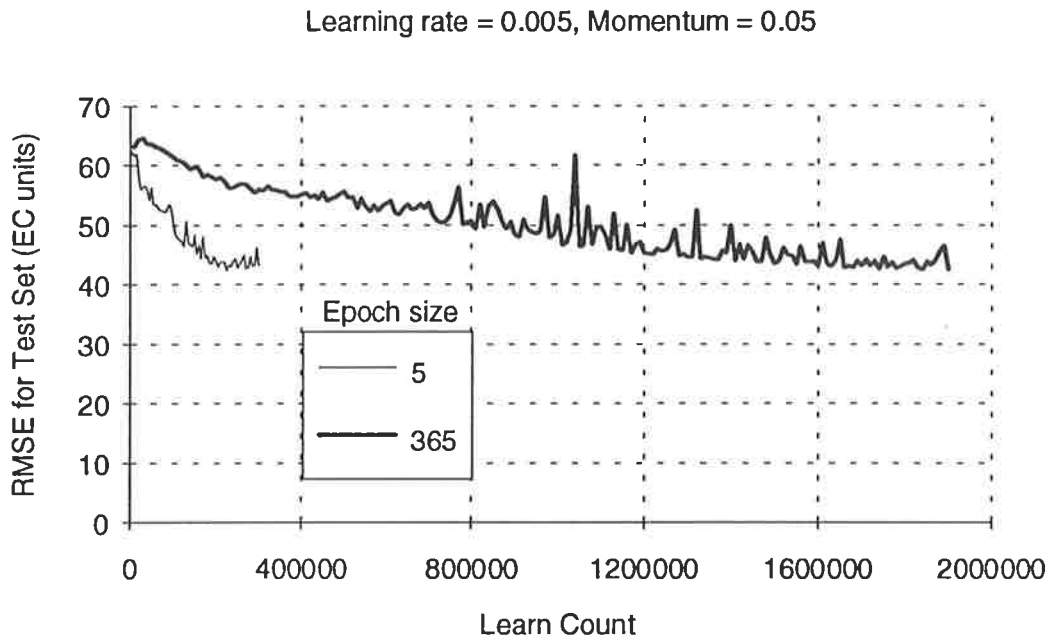


Figure 3.87: RMSE at Different Stages of Learning for Various Epoch Sizes ($\eta = 0.005$, $\mu = 0.05$)

Table 3.44: Best Results for Models With Various Epoch Sizes (η=0.1, μ=0.05)

Model	Epoch size (ε)	RMSE (EC units)	Learn count
SMB_M_A_32_10_88	5	42.3	225,000
SMB_M_A_33_10_88	365	42.4	1,840,000

Momentum and Learning Rate

A number of models were developed (Table 3.45) to assess the effect of learning rate and momentum on network performance. An epoch size of 16 was used in each case. Details of the network parameters and geometries used for the various models are summarised in Appendix D.

Table 3.45: Summary of Models Developed For Assessing the Effects of Different Learning Rates and Momentum Values

Model	Momentum (μ)	Learning rate (η)
SMB_M_A_34_10_88	0.05	0.005
SMB_M_A_38_10_88	0.05	0.02
SMB_M_A_39_10_88	0.05	0.1
SMB_M_A_40_10_88	0.05	0.2
SMB_M_A_35_10_88	0.4	0.005
SMB_M_A_41_10_88	0.4	0.02
SMB_M_A_42_10_88	0.4	0.1
SMB_M_A_43_10_88	0.4	0.2
SMB_M_A_22_10_88	0.6	0.005
SMB_M_A_44_10_88	0.6	0.02
SMB_M_A_30_10_88	0.6	0.1
SMB_M_A_45_10_88	0.6	0.2
SMB_M_A_36_10_88	0.8	0.005
SMB_M_A_46_10_88	0.8	0.02
SMB_M_A_47_10_88	0.8	0.1
SMB_M_A_48_10_88	0.8	0.2
SMB_M_A_37_10_88	0.9	0.005
SMB_M_A_49_10_88	0.9	0.02
SMB_M_A_50_10_88	0.9	0.1

A. Momentum

The inclusion of the momentum term has the effect of adding a proportion of the previous weight change to the current weight change during training (Equation 2.27). A positive momentum value provides a built-in inertia, allowing a small learning rate but faster learning.

Initially, the effect of using momentum values (μ) of 0.05, 0.4, 0.6, 0.8, and 0.9 was investigated at a constant learning rate of 0.005. The RMSEs at the various stages of learning are shown in Figure 3.88 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Table 3.46. It can be seen that for the training set used:

- There is no significant difference between the best forecasts obtained using networks with various momentums.

- The learning speed is much greater for networks using larger momentums.

However, as indicated by the results obtained using model SMB_M_A_37_10_88 (Figure 3.88, $\mu = 0.9$), if large momentums are used, divergent behaviour might occur with continued training.

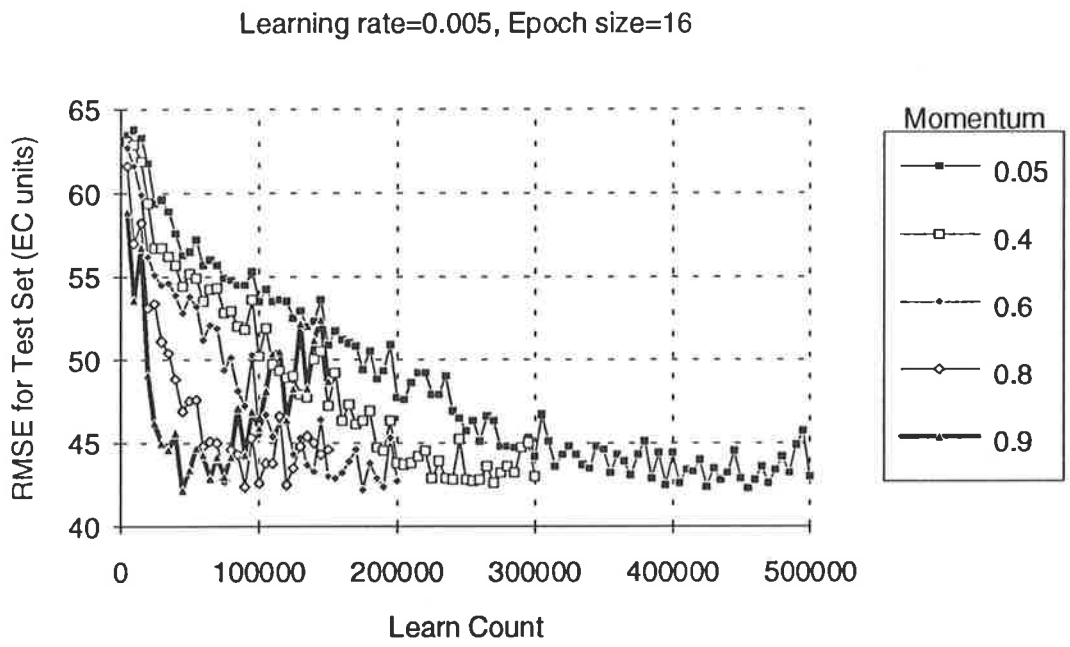


Figure 3.88: RMSE at Different Stages of Learning for Various Values of Momentum ($\eta = 0.005$)

Table 3.46: Best Results for Models With Various Values of Momentum ($\eta = 0.005$, $\varepsilon = 16$)

Model	Momentum (μ)	RMSE (EC units)	Learn count
SMB_M_A_34_10_88	0.05	42.3	455,000
SMB_M_A_35_10_88	0.4	42.6	270,000
SMB_M_A_22_10_88	0.6	42.2	175,000
SMB_M_A_36_10_88	0.8	42.4	90,000
SMB_M_A_37_10_88	0.9	42.2	45,000

The above models (SMB_M_A_34_10_88, SMB_M_A_35_10_88, SMB_M_A_22_10_88, SMB_M_A_36_10_88 and SMB_M_A_37_10_88) were re-trained using different learning rates ($\eta = 0.02$, 0.1 and 0.2), in order to assess whether the effect of different momentum values was the same irrespective of learning rate. The models developed are summarised in Table 3.45.

It should be noted that divergent behaviour occurred during training when a learning rate of 0.1 was used in conjunction with a momentum value of 0.9 (model SMB_M_A_50_10_88) as well as when a learning rate of 0.2 was used in conjunction with a momentum value of 0.8 (model SMB_M_A_48_10_88). This indicates that as the learning rate increases, divergent behaviour starts to occur at lower momentum values.

The RMSEs at the various stages of learning are shown in Figure 3.89 for the models using a learning rate of 0.1 and in Appendix E for all models. The plots for the models using learning rates of 0.02 and 0.2 (not shown here) are very similar to that for the model with a learning rate of 0.1. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Tables 3.47 to 3.49.

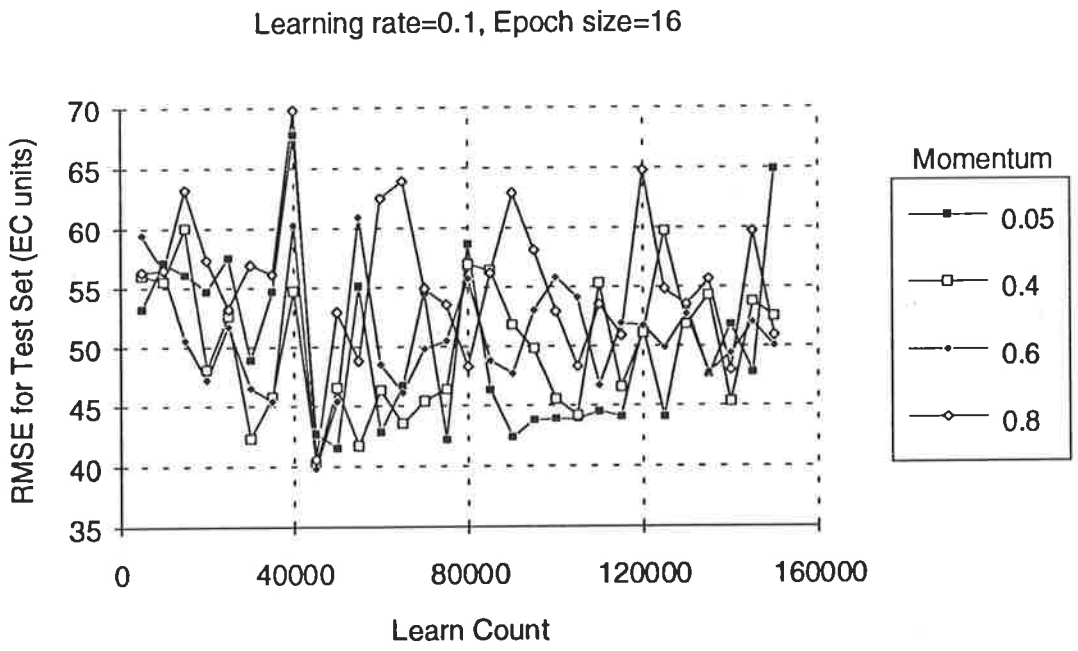


Figure 3.89: RMSE at Different Stages of Learning for Various Values of Momentum ($\eta = 0.1$)

Table 3.47: Best Results for Models With Various Values of Momentum ($\eta = 0.02$, $\varepsilon = 16$)

Model	Momentum (μ)	RMSE (EC units)	Learn count
SMB_M_A_38_10_88	0.05	42.0	115,000
SMB_M_A_41_10_88	0.4	41.9	80,000
SMB_M_A_44_10_88	0.6	42.6	45,000
SMB_M_A_46_10_88	0.8	41.6	45,000
SMB_M_A_49_10_88	0.9	42.4	15,000

Table 3.48: Best Results for Models With Various Values of Momentum ($\eta = 0.1$, $\varepsilon = 16$)

Model	Momentum (μ)	RMSE (EC units)	Learn count
SMB_M_A_39_10_88	0.05	41.5	50,000
SMB_M_A_42_10_88	0.4	40.1	45,000
SMB_M_A_30_10_88	0.6	39.8	45,000
SMB_M_A_47_10_88	0.8	40.6	45,000

Table 3.49: Best Results for Models With Various Values of Momentum ($\eta = 0.2$, $\varepsilon = 16$)

Model	Momentum (μ)	RMSE (EC units)	Learn count
SMB_M_A_40_10_88	0.05	44.1	45,000
SMB_M_A_43_10_88	0.4	39.3	100,000
SMB_M_A_45_10_88	0.6	39.5	30,000

In general the results indicate that the networks learn much faster when higher values of momentum are used, and that the best prediction obtained is independent of momentum. This is in agreement with the results obtained using models SMB_M_A_34_10_88, SMB_M_A_35_10_88, SMB_M_A_22_10_88, SMB_M_A_36_10_88 and SMB_M_A_37_10_88. However, as the learning rate was increased, the behaviour of the networks was less controlled as a result of the large steps taken in weight space. Consequently, the variations in the best RMS prediction errors obtained for the various values of momentum were larger, as indicated in Tables 3.47 to 3.49. In addition, the oscillations of the RMS prediction error were larger, as shown in Figure 3.89.

Models SMB_M_A_34_10_88 and SMB_M_A_22_10_88 were re-trained using epoch sizes of 5 and 365 (rather than 16), in order to assess whether the effect of different momentum values was the same regardless of epoch size. A momentum value of 0.05 was used for model SMB_M_A_32_10_88 and a momentum value of 0.6 was used for model SMB_M_A_20_10_88. An epoch size of 5 was used for both models. A momentum value of 0.05 was used for model SMB_M_A_33_10_88 and a momentum value of 0.6 was used for model SMB_M_A_26_10_88. An epoch size of 365 was used for both models. All other parameters used were the same as in models SMB_M_A_34_10_88 and SMB_M_A_22_10_88. Details of the network parameters and geometries used for the various models are summarised in Appendix D.

The RMSEs at the various stages of learning are shown in Figures 3.90 to 3.92 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Tables 3.50 to 3.52. It can be seen that regardless of epoch size:

- There is no significant difference between the best forecasts obtained using momentums of 0.05 and 0.6.
- The learning speed is much greater when using a larger momentum value.

However, as indicated in Figure 3.88, care must be taken to avoid choosing momentum values that are too large, as divergent behaviour might result.

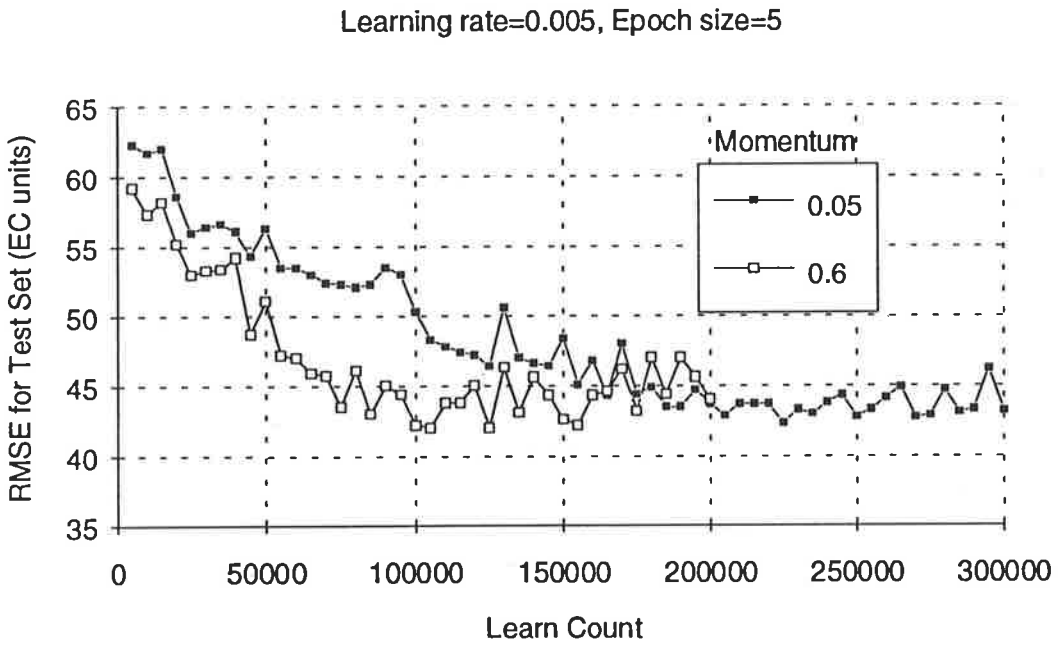


Figure 3.90: RMSE at Different Stages of Learning for Various Values of Momentum ($\varepsilon = 5$)

Table 3.50: Best Results for Models With Various Values of Momentum ($\varepsilon = 5$, $\eta = 0.005$)

Model	Momentum (μ)	RMSE (EC units)	Learn count
SMB_M_A_32_10_88	0.05	42.3	225,000
SMB_M_A_20_10_88	0.6	42.0	105,000

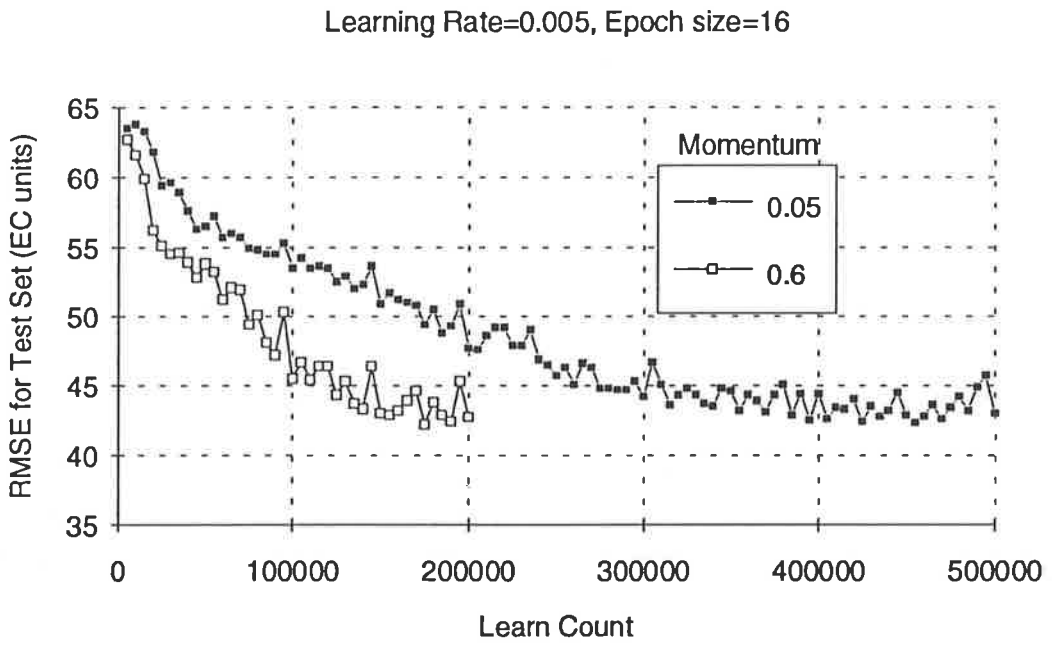


Figure 3.91: RMSE at Different Stages of Learning for Various Values of Momentum ($\varepsilon = 16$)

Table 3.51: Best Results for Models With Various Values of Momentum ($\varepsilon = 16$, $\eta = 0.005$)

Model	Momentum (μ)	RMSE (EC units)	Learn count
SMB_M_A_34_10_88	0.05	42.3	455,000
SMB_M_A_22_10_88	0.6	42.2	175,000

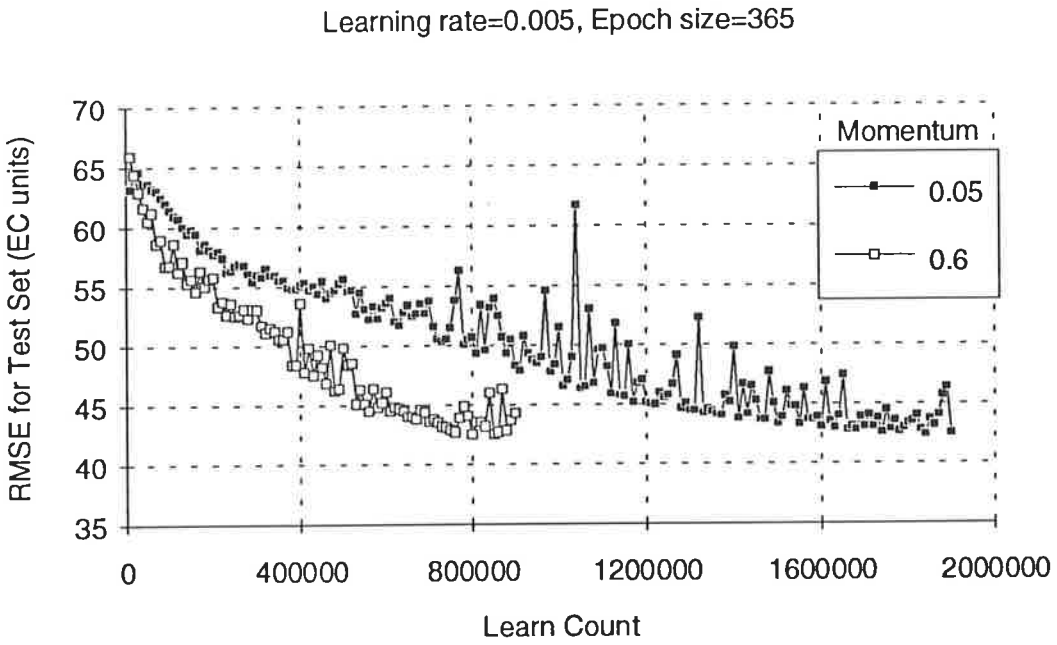


Figure 3.92: RMSE at Different Stages of Learning for Various Values of Momentum ($\varepsilon = 365$)

Table 3.52: Best Results for Models With Various Values of Momentum ($\varepsilon = 365$, $\eta = 0.005$)

Model	Momentum (μ)	RMSE (EC units)	Learn count
SMB_M_A_33_10_88	0.05	42.4	1,840,000
SMB_M_A_26_10_88	0.6	42.5	800,000

B. Learning rate

The amount a particular connection weight is changed is proportional to the learning rate, η (Equation 2.17). The learning rate determines the size of the steps taken in weight space. The theory of back-propagation requires the use of learning rates which approach zero (NeuralWare, Inc., 1991). However, very small learning rates slow down learning, especially on long, shallow portions of the error surface. On the other hand, if the learning rates are too big, the network may go through large oscillations during training or may never converge. Ideally, large learning rates should be used on long, shallow portions of the error surface and smaller values of the learning rate should be used on steep sections and near local minima. However, to select the correct learning rate is difficult. Rumelhart et al. (1986) suggest using learning rates ranging from 0.05 to 0.75, but within this range, the "optimum" learning rate still needs to be determined by trial and error. The same applies to determining at which stage during

training to reduce the learning rate and by how much. Although a lot of research has been carried out on 'optimum' learning rates for a variety of problems, they only serve as a guide, as the effect of different learning rates is highly dependent on the error surface in weight space, which varies from problem to problem.

The effect of using the delta-bar-delta algorithm (Jacobs, 1988) (see Section 2.1.8.5) was also investigated. The delta-bar-delta algorithm assigns individual learning rates to each weight, which are changed in response to changes in the geometry of the error surface as learning progresses. The advantages of this method are that it eliminates the need for guessing the learning rate and increases the convergence speed during training (NeuralWare, Inc., 1991).

Initially, the effect of using learning rates (η) of 0.005, 0.02, 0.1 and 0.2 was investigated at a constant momentum value of 0.05. The RMSEs at the various stages of learning are shown in Figures 3.93 and 3.94 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Table 3.53. It can be seen that:

- For small learning rates, learning progressed slowly and steadily until a local minimum in the error surface was reached.
- As the learning rate was increased, the local minimum in the error surface was reached more rapidly.
- Once the vicinity of a local minimum was reached, oscillations in the RMS prediction error occurred. The magnitude of the oscillations was greater for larger learning rates.
- There is an optimal value of the learning rate ($\eta = 0.1$) in terms of the lowest RMSE reached (Table 3.53). In the vicinity of a local minimum, the slope in the error surface is usually steep, as indicated in Figure 3.95. When the size of the steps taken in weight space is small, networks are unable to escape local minima. For example (Figure 3.95), if the network approaches the local minimum at M1 and is currently at point A, a small step in weight space might result in the network jumping from point A to point B. As training progresses, the network will jump from one side of M1 to the other, and the smallest value that can be obtained is that of M1. On the other hand, if the steps taken in weight space are larger, networks are able to escape local minima and have the opportunity to reach more desirable regions in weight space. For example, if the size of the step taken in weight space in the above example was bigger, the network might jump from point A to point D, leading to a much better result. However, the result obtained is very much dependent on the nature of the error surface (e.g. steepness of slopes, number of local minima), as well as the size of the steps taken in weight space. For example, the above network could just as

easily jump from point A to point C, resulting in an increase in RMSE. Consequently, when larger steps are taken in weight space, the magnitude of the oscillations in RMSE are greater, but so are the chances of obtaining better results. Once the network is in the vicinity of a local minimum, it is basically "pot luck" as to which result is obtained, depending on the point at which training is stopped and the network is tested. Although using larger learning rates enables better solutions to be found, one would expect that more frequent testing of the network needs to be carried out to increase the chances of obtaining a good result.

Care should be taken to ensure that the learning rate used is not too large, as this might result in divergent behaviour.

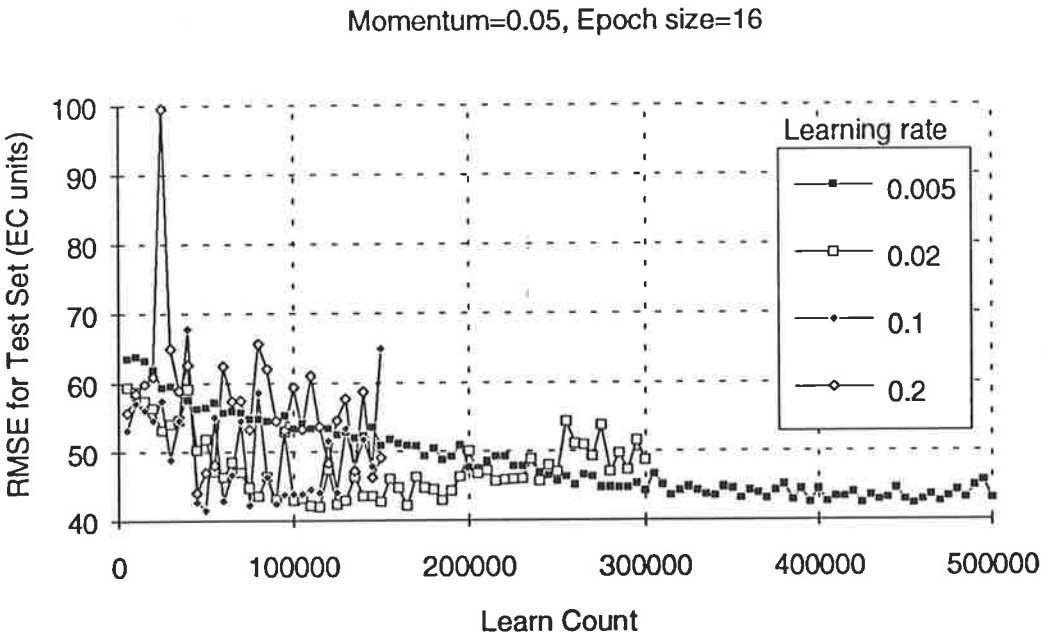


Figure 3.93: RMSE at Different Stages of Learning for Various Learning Rates ($\mu = 0.05$)

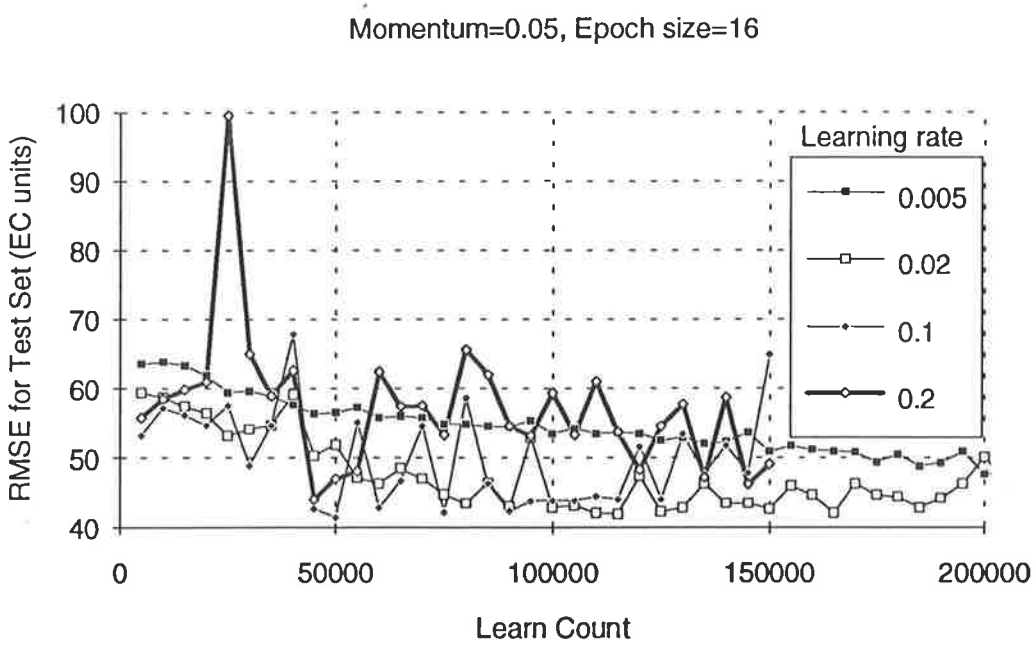


Figure 3.94: RMSE at Different Stages of Learning for Various Learning Rates ($\mu = 0.05$) (Up to a Learn Count of 200,000)

Table 3.53: Best Results for Models With Various Learning Rates ($\mu = 0.05$, $\varepsilon = 16$)

Model	Learning rate (η)	RMSE (EC units)	Learn count
SMB_M_A_34_10_88	0.005	42.3	455,000
SMB_M_A_38_10_88	0.02	42.0	115,000
SMB_M_A_39_10_88	0.1	41.5	50,000
SMB_M_A_40_10_88	0.2	44.1	45,000

The above models (SMB_M_A_34_10_88, SMB_M_A_38_10_88, SMB_M_A_39_10_88 and SMB_M_A_40_10_88) were re-trained using different values of momentum ($\mu = 0.4$, 0.6, 0.8 and 0.9) in order to assess whether the effect of different learning rates is the same regardless of momentum. The models developed are summarised in Table 3.45.

It should be noted that divergent behaviour occurred during training when a momentum value of 0.8 was used in conjunction with a learning rate of 0.2 (model SMB_M_A_48_10_88) as well as when a momentum value of 0.9 was used in conjunction with a learning rate of 0.1 (model SMB_M_A_50_10_88). This indicates that as the momentum value increases, divergent behaviour starts to occur at lower learning rates.

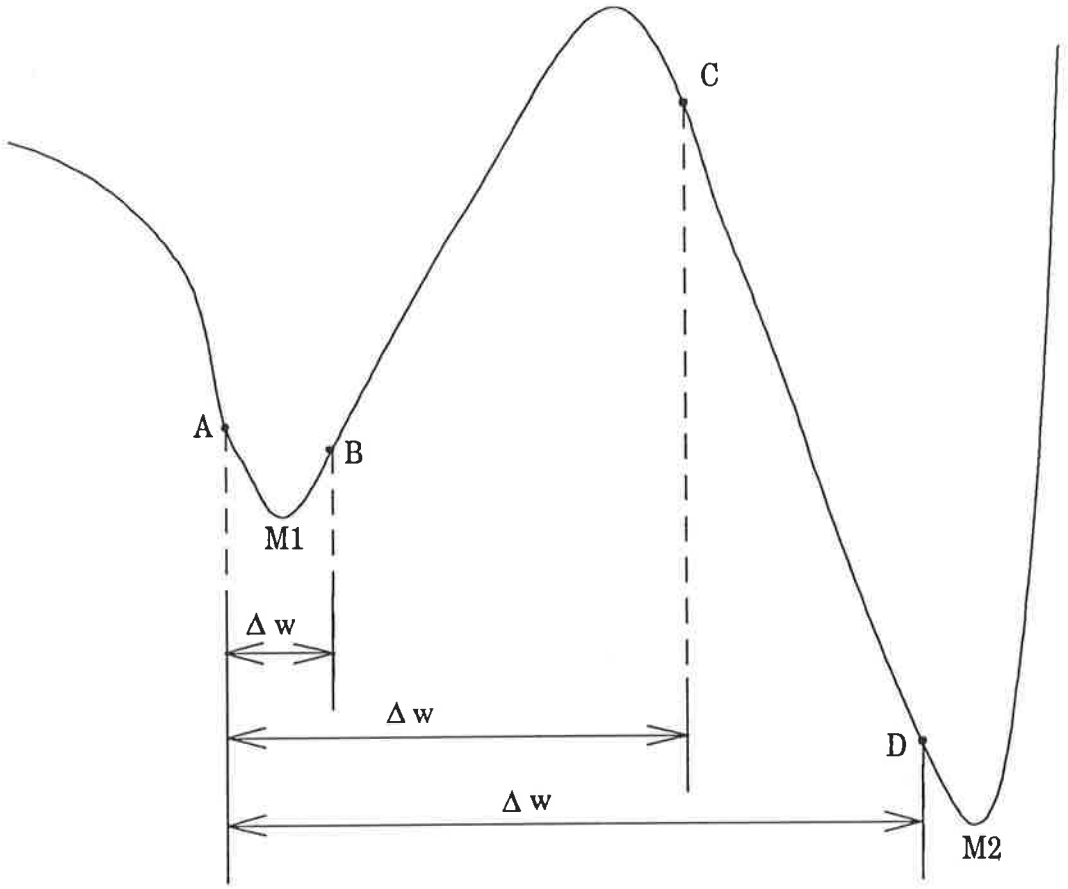


Figure 3.95: Example of an Error Surface in the Vicinity of Local Minima

The RMSEs at the various stages of learning are shown in Figures 3.96 to 3.99 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Tables 3.54 to 3.57. In general, the results indicate that the networks learn much faster when higher learning rates are used and that better predictions are obtained at higher learning rates. This is in agreement with the results obtained when using models SMB_M_A_34_10_88, SMB_M_A_38_10_88, SMB_M_A_39_10_88 and SMB_M_A_40_10_88. It should be noted that divergent behaviour occurred with continued training when the size of the steps taken in weight space was large (i.e. when large learning rates and momentums were used).

The results obtained using model SMB_M_A_43_10_88 (Figure 3.96, learning rate = 0.2 and Appendix E) illustrate that once the network is in the vicinity of a local minimum, it is chance whether a good result is obtained and depends on when training is stopped and the network is tested. As a result of the high learning rate used ($\eta = 0.2$), the vicinity of a local minimum was reached very early (possibly at a learn count of 30,000). Subsequently, large oscillations in the RMSE occurred and, depending on when the network was tested, a best prediction RMSE of 68.8 EC units or 39.3 EC units could have been obtained.

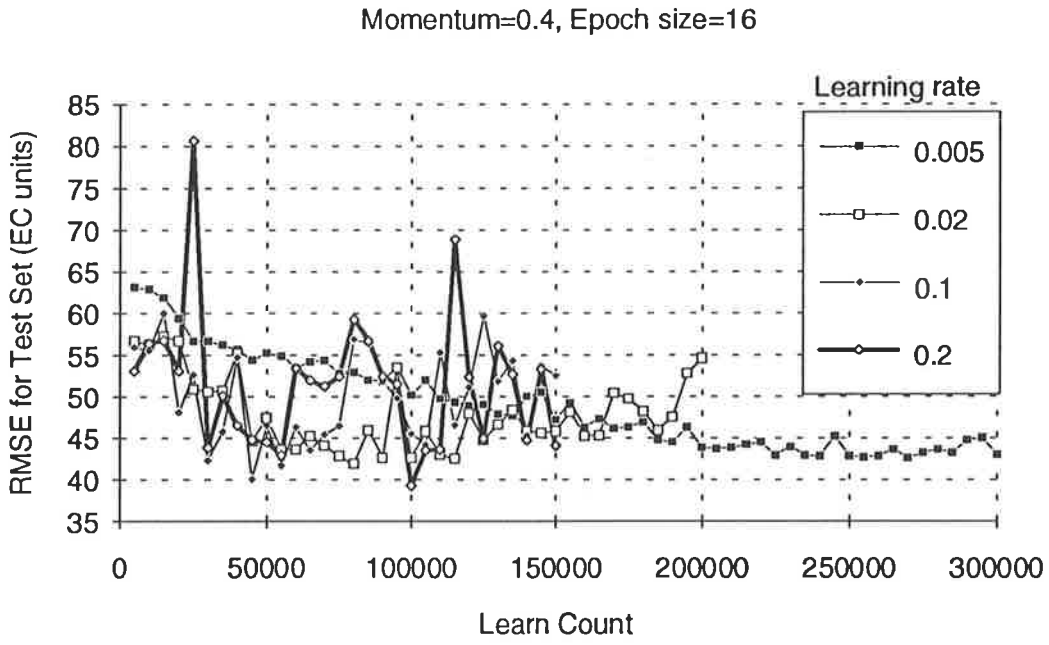


Figure 3.96: RMSE at Different Stages of Learning for Various Learning Rates ($\mu = 0.4$)

Table 3.54: Best Results When Using Various Learning Rates ($\mu = 0.4$, $\varepsilon = 16$)

Model	Learning rate (η)	RMSE (EC units)	Learn count
SMB_M_A_35_10_88	0.005	42.6	270,000
SMB_M_A_41_10_88	0.02	41.9	80,000
SMB_M_A_42_10_88	0.1	40.1	45,000
SMB_M_A_43_10_88	0.2	39.3	100,000

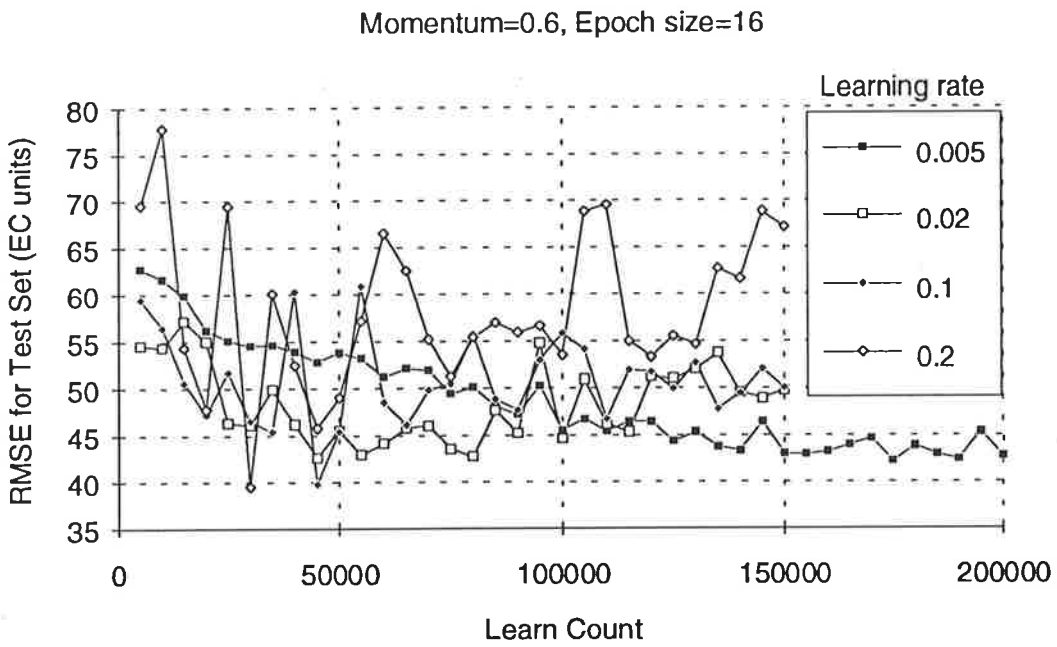


Figure 3.97: RMSE at Different Stages of Learning for Various Learning Rates ($\mu = 0.6$)

Table 3.55: Best Results When Using Various Learning Rates ($\mu = 0.6$, $\varepsilon = 16$)

Model	Learning rate (η)	RMSE (EC units)	Learn count
SMB_M_A_22_10_88	0.005	42.2	175,000
SMB_M_A_44_10_88	0.02	42.6	45,000
SMB_M_A_30_10_88	0.1	39.8	45,000
SMB_M_A_45_10_88	0.2	39.5	30,000

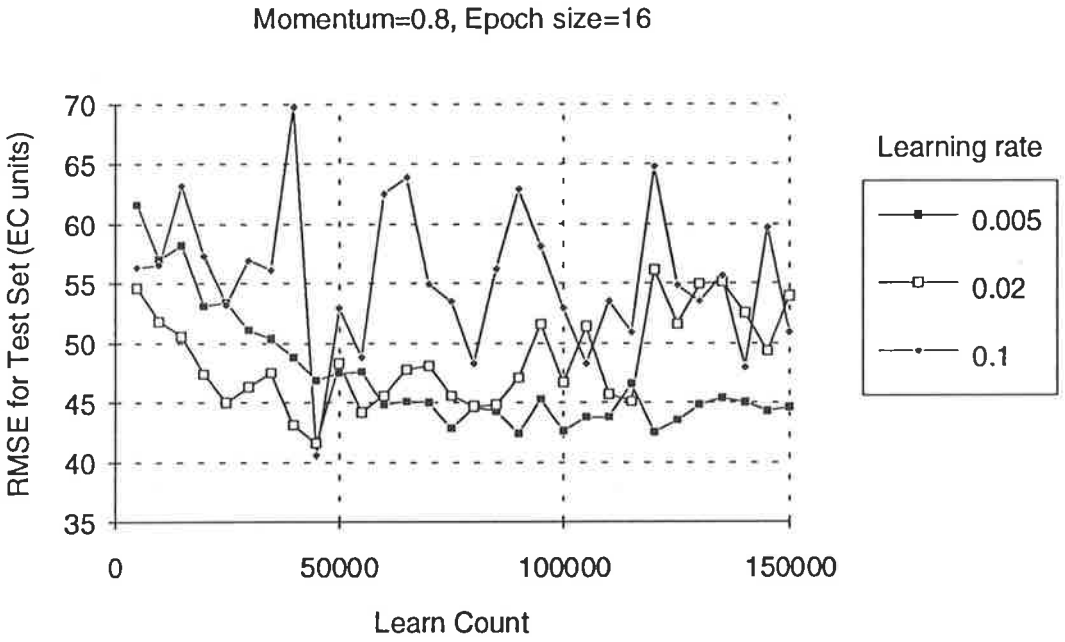


Figure 3.98: RMSE at Different Stages of Learning for Various Learning Rates ($\mu = 0.8$)

Table 3.56: Best Results When Using Various Learning Rates ($\mu = 0.8$, $\varepsilon = 16$)

Model	Learning rate (η)	RMSE (EC units)	Learn count
SMB_M_A_36_10_88	0.005	42.4	90,000
SMB_M_A_46_10_88	0.02	41.6	45,000
SMB_M_A_47_10_88	0.1	40.6	45,000

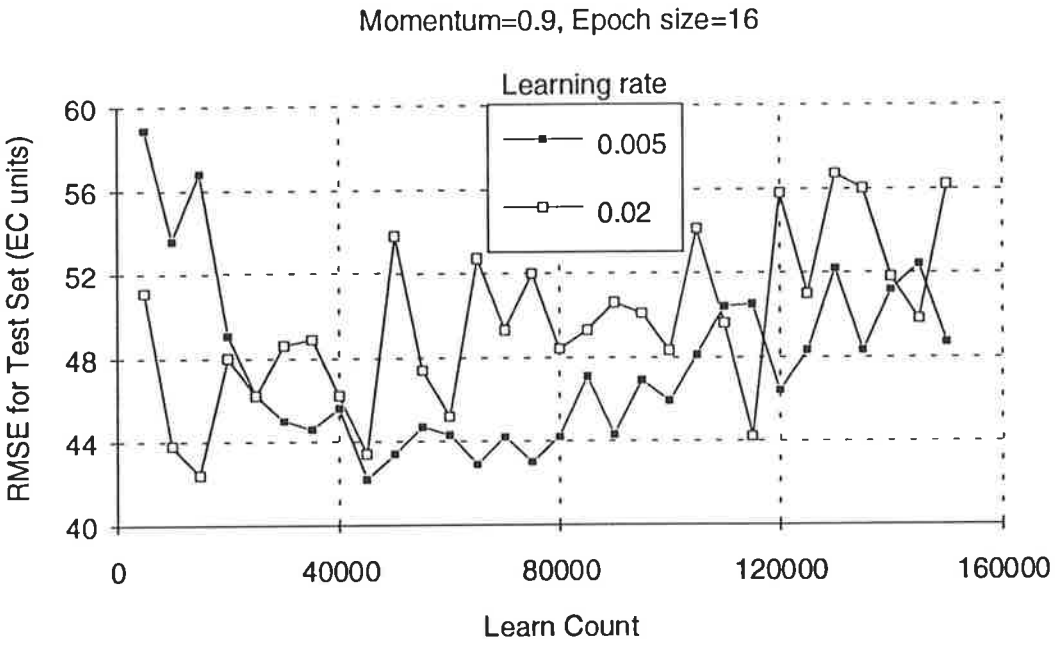


Figure 3.99: RMSE at Different Stages of Learning for Various Learning Rates ($\mu = 0.9$)

Table 3.57: Best Results When Using Various Learning Rates ($\mu = 0.9$, $\varepsilon = 16$)

Model	Learning rate (η)	RMSE (EC units)	Learn count
SMB_M_A_37_10_88	0.005	42.2	45,000
SMB_M_A_49_10_88	0.02	42.4	15,000

Models SMB_M_A_22_10_88 and SMB_M_A_30_10_88 were re-trained using epoch sizes of 5 and 365 (rather than 16) in order to assess whether the effect of different learning rates is the same regardless of epoch size. A learning rate of 0.005 was used for model SMB_M_A_20_10_88 and a learning rate of 0.1 was used for model SMB_M_A_28_10_88. An epoch size of 5 was used for both models. A learning of 0.005 was used for model SMB_M_A_21_10_88 and a learning rate of 0.1 was used for model SMB_M_A_29_10_88. An epoch size of 10 was used for both models. A learning rate of 0.005 was used for model SMB_M_A_23_10_88 and a learning rate of 0.1 was used for model SMB_M_A_31_10_88. An epoch size of 35 was used for both models. All other parameters used were the same as in models SMB_M_A_22_10_88 and SMB_M_A_30_10_88. Details of the network parameters and geometries used for the various models are summarised in Appendix D.

The RMSEs at the various stages of learning are shown in Figures 3.100 to 3.103 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they

were obtained are shown in Tables 3.58 to 3.61. It can be seen that regardless of epoch size:

- Using larger learning rates generally results in better predictions (but the result obtained is highly dependent on when the network is tested as shown when using model SMB_M_A_31_10_88).
- Learning speed is much greater when using larger learning rates.
- Divergent behaviour occurs if training is continued when using the larger learning rate (i.e. $\eta = 0.1$).

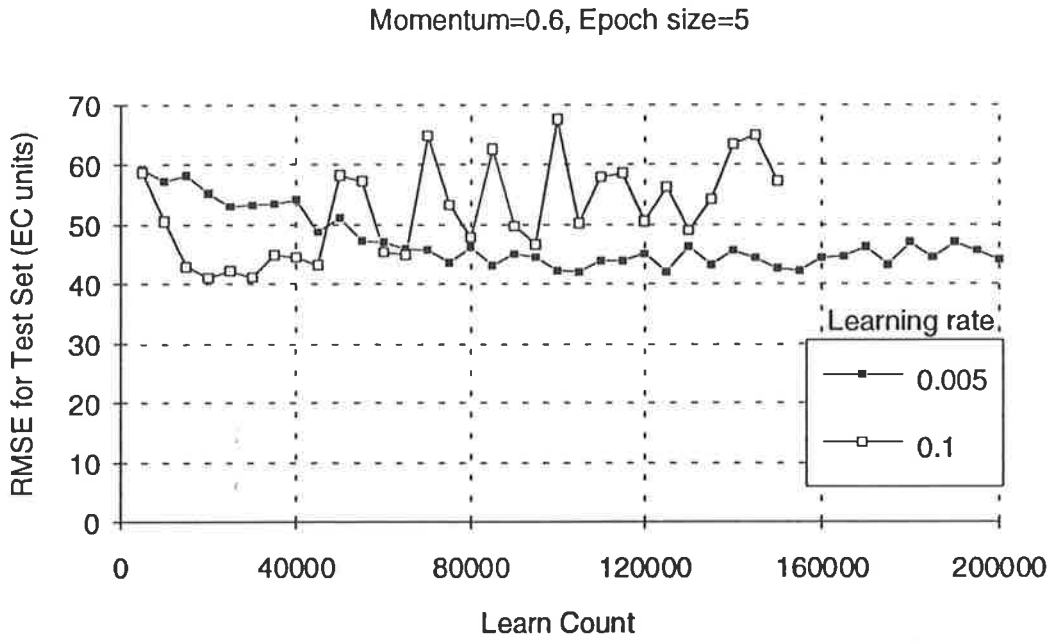


Figure 3.100: RMSE at Different Stages of Learning for Various Learning Rates ($\varepsilon = 5$)

Table 3.58: Best Results When Using Various Learning Rates ($\varepsilon = 5$, $\mu = 0.6$)

Model	Learning rate (η)	RMSE (EC units)	Learn count
SMB_M_A_20_10_88	0.005	42.0	105,000
SMB_M_A_28_10_88	0.1	41.0	20,000

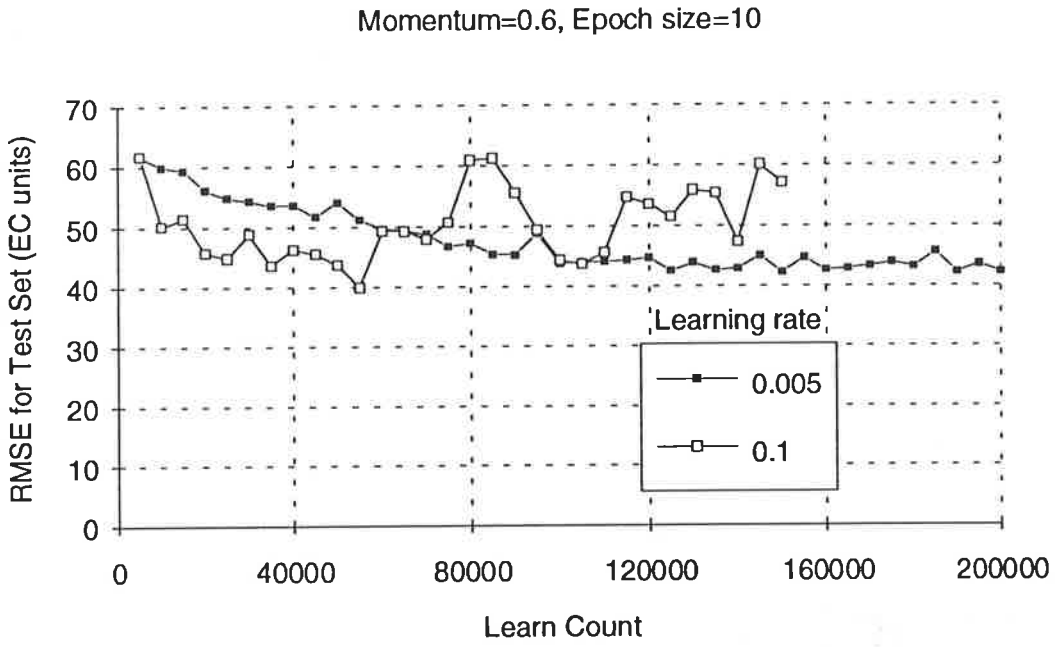


Figure 3.101: RMSE at Different Stages of Learning for Various Learning Rates ($\varepsilon = 10$)

Table 3.59: Best Results When Using Various Learning Rates ($\varepsilon = 10$, $\mu = 0.6$)

Model	Learning rate (η)	RMSE (EC units)	Learn count
SMB_M_A_21_10_88	0.005	42.2	150,000
SMB_M_A_29_10_88	0.1	39.9	55,000

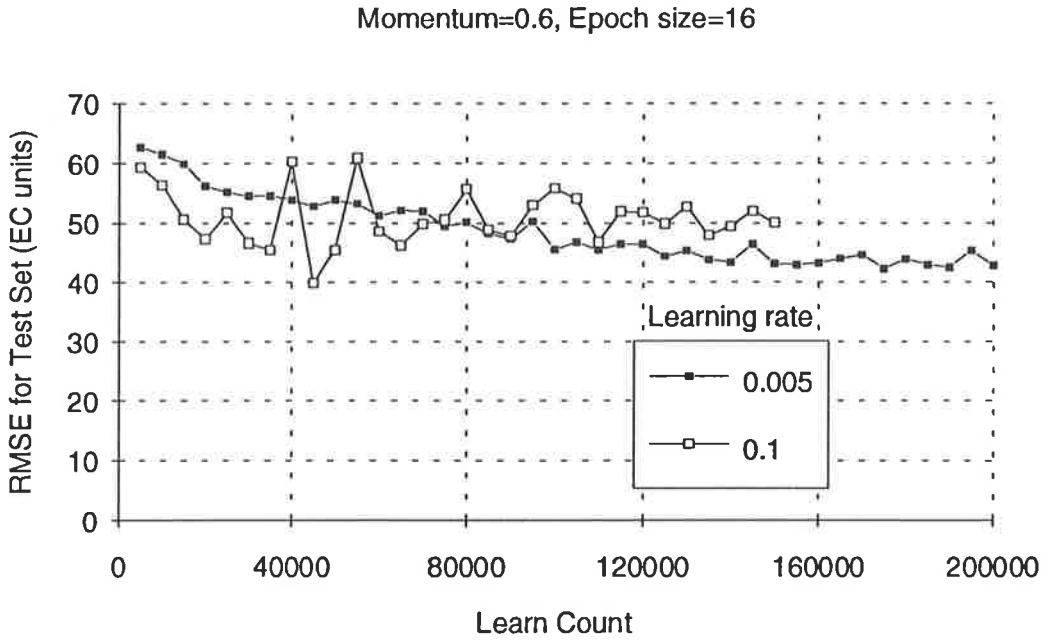


Figure 3.102: RMSE at Different Stages of Learning for Various Learning Rates ($\varepsilon = 16$)

Table 3.60: Best Results When Using Various Learning Rates ($\varepsilon = 16$, $\mu = 0.6$)

Model	Learning rate (η)	RMSE (EC units)	Learn count
SMB_M_A_22_10_88	0.005	42.2	175,000
SMB_M_A_30_10_88	0.1	39.8	45,000

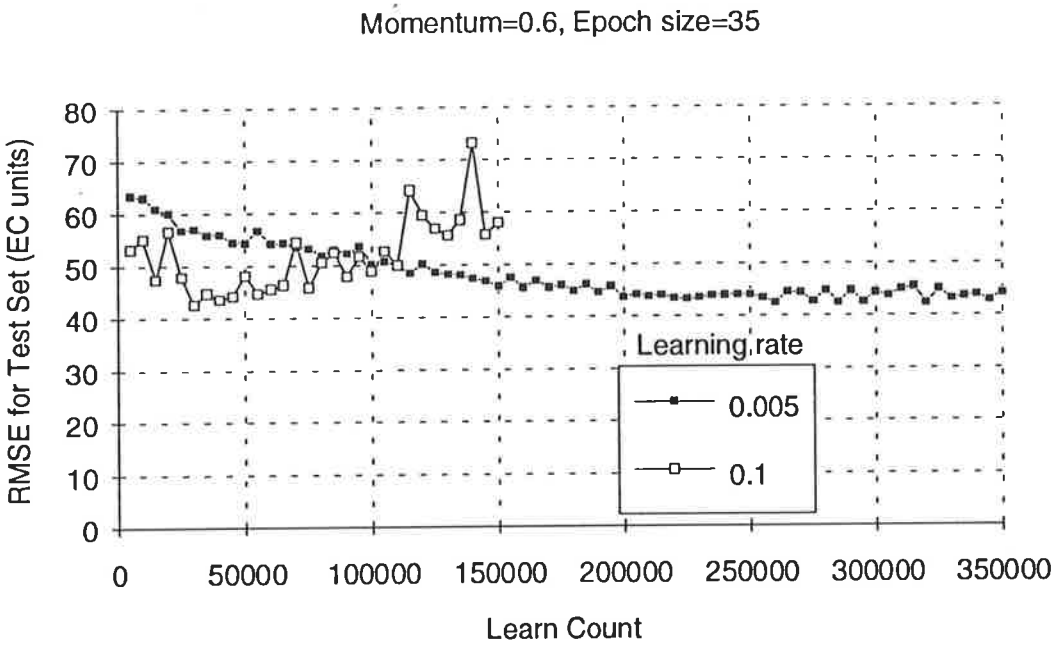


Figure 3.103: RMSE at Different Stages of Learning for Various Learning Rates ($\varepsilon = 35$)

Table 3.61: Best Results When Using Various Learning Rates ($\varepsilon = 35$, $\mu = 0.6$)

Model	Learning rate (η)	RMSE (EC units)	Learn count
SMB_M_A_23_10_88	0.005	42.3	320,000
SMB_M_A_31_10_88	0.1	42.6	30,000

B1. Effect of using the delta-bar-delta algorithm

When using the delta-bar-delta algorithm as part of NeuralWorks Professional II/Plus (NeuralWare, Inc., 1991), two parameters have to be chosen, which determine the rate of adjustment of the learning rates in response to changes in the geometry of the error surface. In model SMB_M_A_51_10_88, the parameters were chosen so that the learning rates were adjusted rapidly in response to changes in the geometry of the error surface and in model SMB_M_A_52_10_88, the parameters were chosen so that the changes in the learning rates in response to changes in the geometry of the error surface were quite small. The results obtained were compared with those obtained using model SMB_M_A_20_10_88, which uses a constant learning rate of 0.005 and a momentum value of 0.6. Details of the network parameters and geometries used for the various models are summarised in Appendix D.

The RMSEs at the various stages of learning are shown in Figure 3.104 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Table 3.62. It can be seen that for the training set used:

- The results obtained are very similar, suggesting that using the delta-bar-delta algorithm does not improve the predictive ability of the network.
- When using the delta-bar-delta algorithm, training speed varies considerably depending on the size of the parameters chosen that control the rate of adjustment of the learning rates. The effect of choosing different values of these parameters appears to be the same as choosing different learning rates. Consequently there is no advantage in using the delta-bar-delta algorithm. Although the delta-bar-delta algorithm eliminates the need for choosing the learning rate, it requires other parameters to be chosen, which appear to have a similar effect.

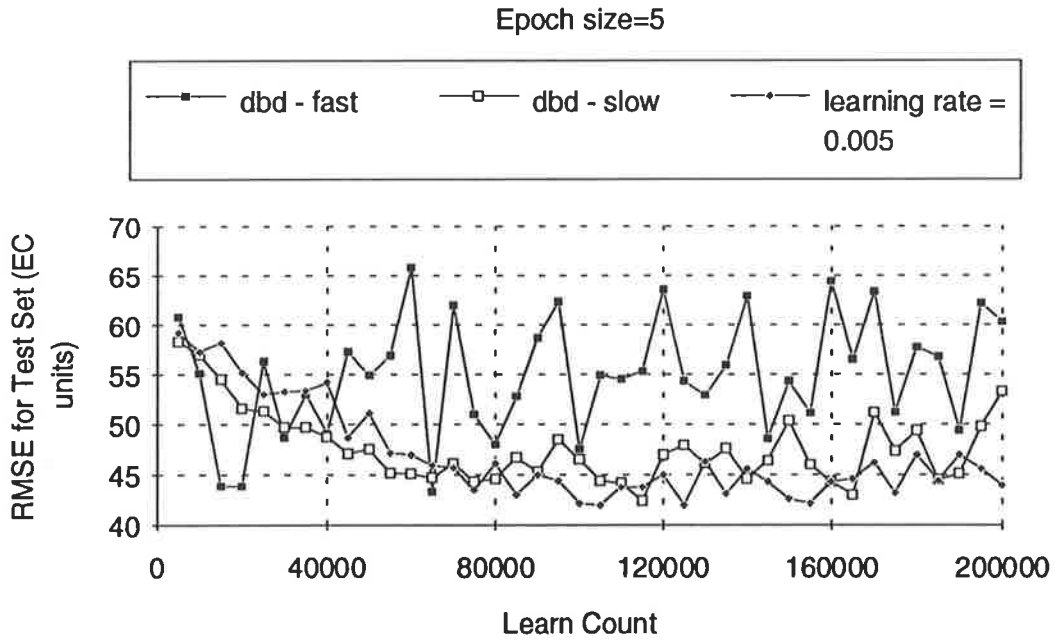


Figure 3.104: RMSE at Different Stages of Learning for Models Using a Learning Rate of 0.005 and the Delta-Bar-Delta Algorithm

Table 3.62: Comparison of the Results Obtained Using a Learning Rate of 0.005 and the Delta-Bar-Delta Algorithm (dbd)

Model	Learning rate (η)	RMSE (EC units)	Learn count
SMB_M_A_20_10_88	0.005	42.0	105,000
SMB_M_A_51_10_88	dbd (slow)	42.4	115,000
SMB_M_A_52_10_88	dbd (fast)	43.3	65,000

Transfer function

The transfer function is a mathematical formula that gives the output of a processing element as a function of its input signal. The effect of using the hyperbolic tangent (Figure 2.7), sigmoidal (Figure 2.6) as well as the linear transfer function (Figure 2.5) was assessed in this series of tests.

The hyperbolic tangent transfer function was used in model SMB_M_A_20_10_88, the sigmoidal transfer function was used in model SMB_M_A_53_10_88 and the linear transfer function was used in model SMB_M_A_54_10_88. Details of the network parameters and geometries used for the various models are summarised in Appendix D.

Training using the model with the sigmoidal transfer function was extremely slow. Consequently, the learning rate used for model SMB_M_A_53_10_88 was increased from 0.005 to 0.1 in order to speed up the learning process. It should be noted that the size of the steps taken in weight space during training, and hence learning speed, is proportional to the derivative of the transfer function (Equation 2.16). If the gain (the slope of the linearly varying portion of the transfer function) of the transfer function is larger, the derivative of the transfer function, and hence the steps taken in weight space, are larger, resulting in increased training speed (Figure 2.11). By inspecting Figures 2.6 and 2.7, it can be seen that the gain of the hyperbolic tangent transfer function is greater than that of the sigmoidal transfer function. As a result, one would expect training to be slower when using the sigmoidal transfer function. The software used does not cater for alterations in the gain of the transfer functions and hence a greater learning rate was used to speed up training when using the sigmoidal transfer function.

The RMSEs at the various stages of learning are shown in Figure 3.105 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Table 3.63. It can be seen that for the training set used:

- There is a considerable difference in learning speed when the various transfer functions were used. The network using the hyperbolic tangent transfer function learned most quickly. The network using the linear transfer function took about twice as long to obtain its best result. Learning with the network using the sigmoidal transfer function was extremely slow, even when a considerably larger learning rate was used. However, as discussed earlier, this is a result of the different gains of the various transfer functions and can be compensated for.
- The predictions obtained using networks with the hyperbolic tangent and sigmoidal transfer functions were comparable, although slightly better results were obtained when the hyperbolic tangent transfer function was used. The best result obtained

using the linear transfer function, on the other hand, was considerably worse than the other two results, suggesting that there is a distinct advantage in using non-linear transfer functions.

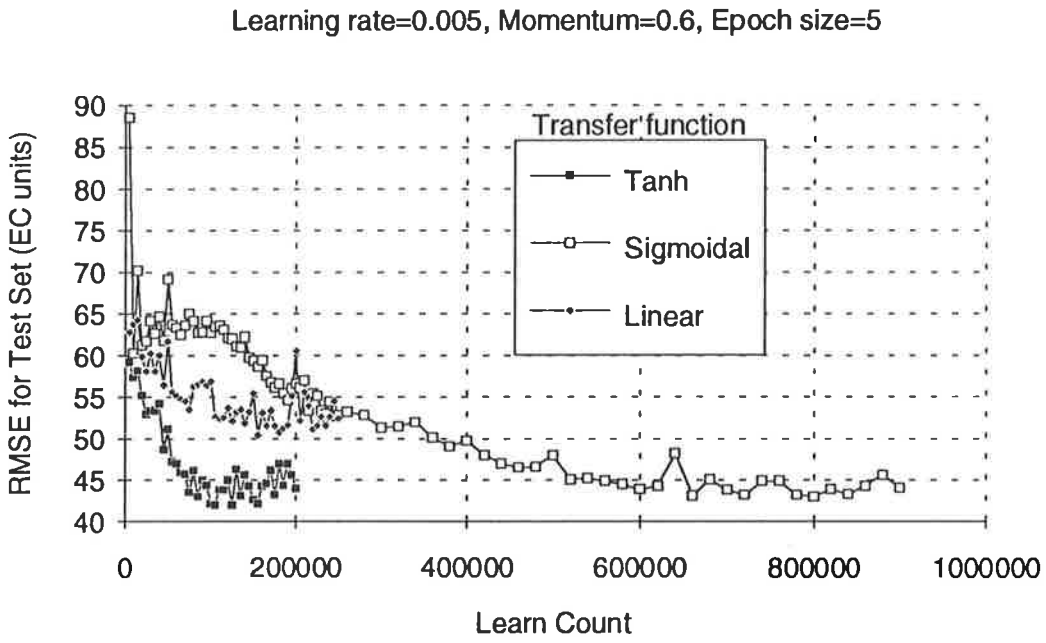


Figure 3.105: RMSE at Different Stages of Learning for Models with Different Transfer Functions

Table 3.63: Best Results Using Various Transfer Functions

Model	Transfer function	RMSE (EC units)	Learn count
SMB_M_A_20_10_88	tanh	42.0	105,000
SMB_M_A_53_10_88	sigmoidal	43.1	660,000
SMB_M_A_54_10_88	linear	51.1	220,000

Error function

During training, each weight is changed in proportion to the size and direction of the negative gradient of the error surface. This gradient is the partial derivative of the global error function with respect to the weight considered. Many different global error functions can be used. NeuralWorks Professional II/Plus caters for the use of the quadratic error function (Equation 2.15) as well as the use of cubic and quartic error functions, given in Equations 2.29 and 2.30 respectively. The quadratic error function was used in model SMB_M_A_20_10_88, the cubic error function was used in model SMB_M_A_55_10_88 and the quartic error function was used in model

SMB_M_A_56_10_88. Details of the network parameters and geometries used for the various models are summarised in Appendix D.

The RMSEs at the various stages of learning are shown in Figure 3.106 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Table 3.64. It can be seen that for the training set used, there is a considerable difference in learning speed and predictive ability, depending on the error function used:

- Using the quadratic error function resulted in the fastest learning speed as well as the best predictive ability.
- Using the cubic error function resulted in the slowest learning speed as well as the worst predictive ability.
- The performance of the network using the quartic transfer function was in between that of the networks using the standard and the cubic error functions.

Table 3.64: Best Results Using Various Error Functions

Model	Transfer function	RMSE (EC units)	Learn count
SMB_M_A_20_10_88	standard	42.0	105,000
SMB_M_A_55_10_88	cubic	49.5	1,240,000
SMB_M_A_56_10_88	quartic	45.4	400,000

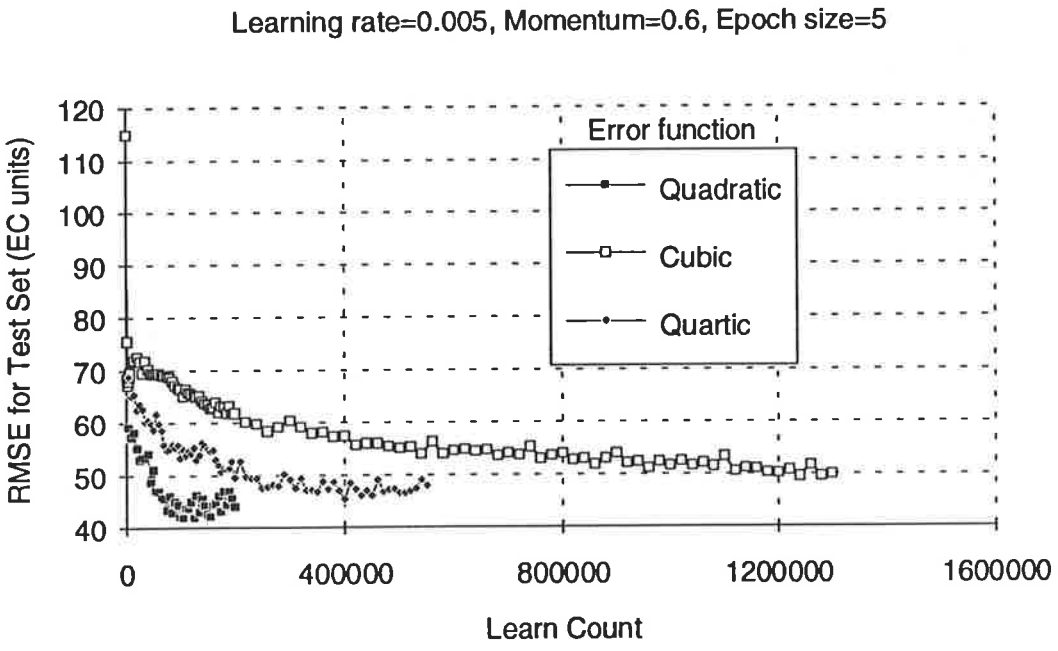


Figure 3.106: RMSE at Different Stages of Learning for Models with Different Error Functions

Initial weight distribution

If training is commenced from an unfavourable position in the weight space, the network may get stuck in a local minimum, from which it cannot escape. By starting at a different (possibly more favourable) position in the weight space (i.e. different initial random weights), a different local minimum may be reached.

Model SMB_M_A_20_10_88 was re-trained twice (models SMB_M_A_20a_10_88 and SMB_M_A_20b_10_88), starting from different positions in weight space each time. The random initial weights were obtained with the aid of NeuralWorks Explorer's "Initialise" function. The upper and lower limits for the initial weights were +0.1 and -0.1.

The results in Table 3.65 show that for the training set used, commencing training from different points in weight space had no significant effect on learning speed and network performance.

Table 3.65: Best Results Using Different Initial Weight Distributions

Model	RMSE (EC units)	Learn count
SMB_M_A_20_10_88	42.2	100,000
SMB_M_A_20a_10_88	42.3	90,000
SMB_M_A_20b_10_88	42.7	90,000

3.5.3.4 Effect of Geometry on Network Performance

The function of the hidden layer nodes is to detect relationships between network inputs and outputs. If there is an insufficient number of hidden nodes, it may be difficult to obtain convergence during training, as the network may be unable to create adequately complex decision boundaries. On the other hand, if too many hidden nodes are used, the network may lose its ability to generalise. In addition, keeping the number of hidden layer nodes to a minimum reduces the number of weights that need to be adjusted and hence reduces the computational time needed for training.

In this section, the effect of using one and two hidden layers was investigated. In the case of networks with two hidden layers, the effect of using different ratios of first to second hidden layer nodes, as well as the effect of using different numbers of nodes in each hidden layer, while maintaining a constant ratio between them, was considered. In

the case of networks with one hidden layer, the effect of the number of nodes in the hidden layer was investigated.

Ratio of first to second hidden layer nodes

In this series of tests, the effect of using different ratios of first to second hidden layer nodes was assessed. The number of nodes in the first hidden layer was kept constant at 45. The number of nodes used in the second hidden layer include 5 (model SMB_M_A_57_10_88), 10 (model SMB_M_A_58_10_88), 15 (model SMB_M_A_20_10_88), 20 (model SMB_M_A_59_10_88) and 25 (model SMB_M_A_60_10_88). Details of the network parameters and geometries used for the various models are summarised in Appendix D.

The RMSEs at the various stages of learning are shown in Figure 3.107 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Table 3.66. It can be seen that for the training set used, learning speed and generalisation ability do not appear to be significantly affected by the ratio of first to second hidden layer nodes.

The results in Table 3.66 indicate that there is some variation in the generalisation ability of the various models. However, there is no clear trend, although there is a minimum in the RMS prediction error when 15 nodes were used in the second hidden layer. This corresponds to a ratio of 3:1 between the first and second hidden layer nodes and is in agreement with the results obtained by Kudrycki (1988). It should be noted, though, that this result is by no means conclusive, as only one test was carried out and the small variation of RMS prediction error obtained for the various tests could be due to other factors such as testing frequency (Section 3.5.3.3).

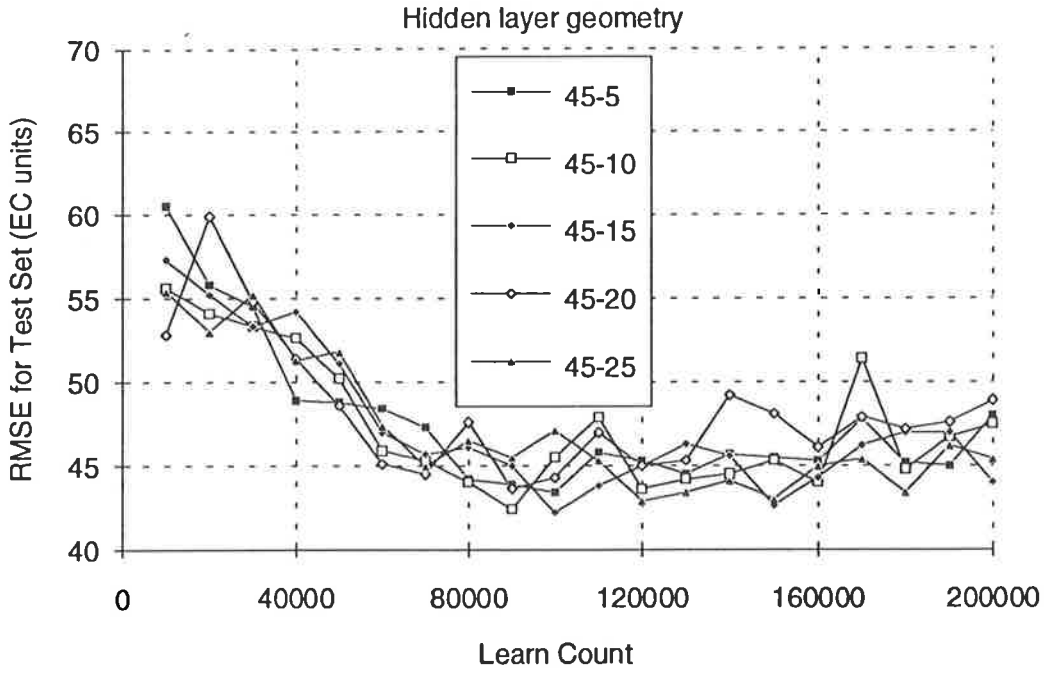


Figure 3.107: RMSE at Various Stages of Learning for Different Ratios of First to Second Hidden Layer Nodes

Table 3.66: Best Results Using Different Ratios of First to Second Hidden Layer Nodes

Model	N^{H1}	N^{H2}	Ratio $N^{H1}:N^{H2}$	RMSE (EC units)	Learn count
SMB_M_A_57_10_88	45	5	9:1	43.4	100,000
SMB_M_A_58_10_88	45	10	4.5:1	42.4	90,000
SMB_M_A_20_10_88	45	15	3:1	42.2	100,000
SMB_M_A_59_10_88	45	20	2.25:1	43.6	90,000
SMB_M_A_60_10_88	45	25	1.8:1	42.9	120,000

Number of nodes in the first and second hidden layers

In this series of tests, a ratio of first to second hidden layer nodes of 3:1 was maintained, while varying the number of nodes in the first hidden layer (and consequently in the second hidden layer). The number of nodes used in the first hidden layer was 15 (model SMB_M_A_61_10_88), 30 (model SMB_M_A_62_10_88), 45 (model SMB_M_A_20_10_88), 60 (model SMB_M_A_63_10_88) and 90 (model SMB_M_A_64_10_88). Details of the network parameters and geometries used for the various models are summarised in Appendix D.

The RMSEs at the various stages of learning are shown in Figure 3.108 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Table 3.67. It can be seen that for the training set used:

- The best results were obtained at approximately the same learn count for all tests. However, as the time taken to perform one weight update is considerably greater for larger networks, there is quite a large variation in training speed.
- There is some difference in the generalisation ability of the networks. There appears to be an increase in generalisation ability with an increase in the number of hidden layer nodes up to a certain point, with optimum performance being achieved when 60 nodes were used in the first hidden layer (Table 3.67). It should be noted, however, that this result is by no means conclusive, as only one test was carried out and the variation of RMS prediction error obtained for the various tests could be due to other factors such as testing frequency (Section 3.5.3.3).

Table 3.67: Best Results Using Different Numbers of Hidden Layer Nodes

Model	N^{H1} - N^{H2}	RMSE (EC units)	Learn count
SMB_M_A_61_10_88	15-5	44.9	130,000
SMB_M_A_62_10_88	30-10	44.0	90,000
SMB_M_A_20_10_88	45-15	42.2	100,000
SMB_M_A_63_10_88	60-20	41.9	90,000
SMB_M_A_64_10_88	90-30	42.4	110,000

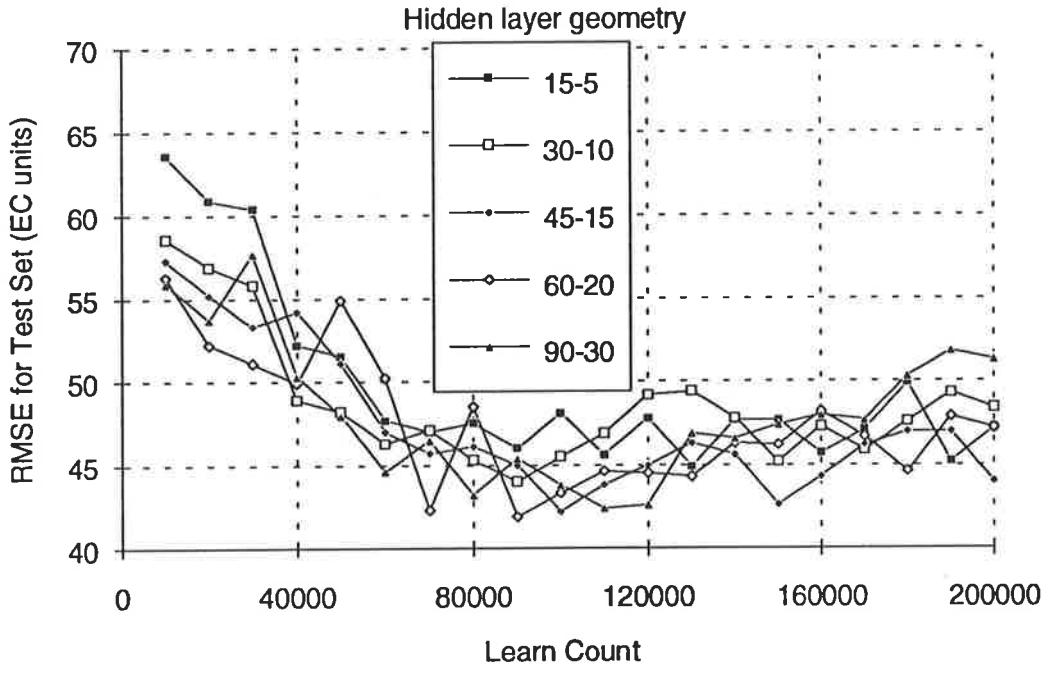


Figure 3.108: RMSE at Various Stages of Learning for Different Numbers of Nodes in the Hidden Layers

Number of nodes in the first hidden layer

In this series of tests only one layer of hidden nodes was used. The effect of the number of nodes in the hidden layer on network performance was investigated. The number of nodes used in the hidden layer include 15 (model SMB_M_A_65_10_88), 30 (model SMB_M_A_66_10_88), 45 (model SMB_M_A_67_10_88), 60 (model SMB_M_A_68_10_88) and 90 (model SMB_M_A_69_10_88). Details of the network parameters and geometries used for the various models are summarised in Appendix D.

The RMSEs at the various stages of learning are shown in Figure 3.109 and in Appendix E. The RMSEs of the best 14 day forecasts and the learn count at which they were obtained are shown in Table 3.68. It can be seen that for the training set used:

- The best results were obtained at approximately the same learn count for all models. However, as the time taken to perform one weight update is considerably greater for larger networks, there is quite a large variation in training speed.
- There is some variation in the generalisation ability of the networks. However, there appears to be no trend. The best result was obtained when 30 hidden layer nodes were used (Table 3.68). It should be noted, though, that this result is by no means conclusive, as only one test was carried out and the variation of RMS prediction error obtained for the various tests could be due to other factors such as testing frequency (Section 3.5.3.3).

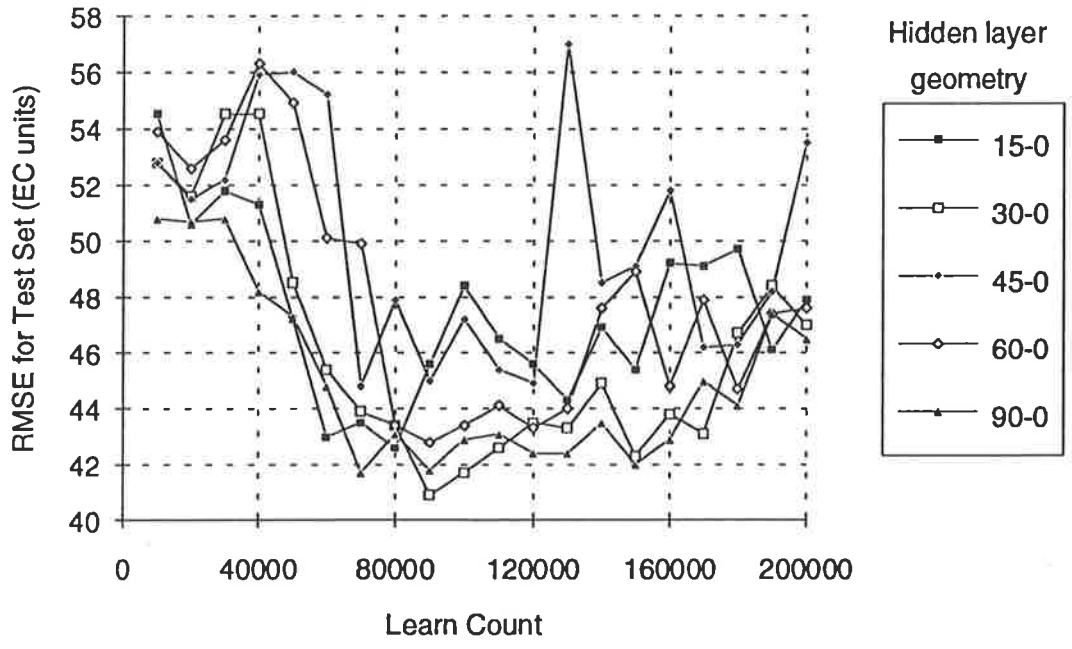


Figure 3.109: RMSE at Various Stages of Learning for Different Numbers of Nodes in the First Hidden Layer

Table 3.68: Best Results Using Different Numbers of Nodes in the First Hidden Layer

Model	N^{H1} - N^{H2}	RMSE (EC units)	Learn count
SMB_M_A_65_10_88	15-0	42.6	80,000
SMB_M_A_66_10_88	30-0	40.9	90,000
SMB_M_A_67_10_88	45-0	44.8	70,000
SMB_M_A_68_10_88	60-0	42.8	90,000
SMB_M_A_69_10_88	90-0	41.7	70,000

3.5.3.5 Application of Results

The "optimum" network parameters and network geometries determined in Sections 3.5.3.3 and 3.5.3.4 were used to train a number of networks in an attempt to improve the 14 day forecast of salinity in the River Murray at Murray Bridge obtained for 1988 in Section 3.5.2. The network parameters that were found to be optimal in Section 3.5.3.3 are summarised in Table 3.69.

Table 3.69: Optimum Network Parameters

Parameter	Optimum value
Learning rate	0.1
Momentum	0.6
Epoch size	5
Transfer function	Hyperbolic Tangent
Error function	Quadratic

The parameters listed in Table 3.69 were used to train a network with one hidden layer consisting of 30 nodes (model SMB_M_A_70_10_88), as this was the geometry that produced the best forecast in Section 3.5.3.4 (model SMB_M_A_66_10_88). It should be noted that model SMB_M_A_70_10_88 is identical to model SMB_M_A_66_10_88, with the exception of an increase in the learning rate from 0.005 to 0.1.

A second network was also trained. The parameters shown in Table 3.69 were used in conjunction with 60 nodes in the first hidden layer and 20 nodes in the second hidden layer (model SMB_M_A_71_10_88), as this was the geometry that produced the best forecasts for networks with 2 hidden layers (model SMB_M_A_63_10_88). It should

be noted that model SMB_M_A_71_10_88 is identical to model SMB_M_A_63_10_88, with the exception of an increase in the learning rate from 0.005 to 0.1. A summary of the network parameters and geometries used for models SMB_M_A_70_10_88 and SMB_M_A_71_10_88 is provided in Appendix D.

The results obtained are shown in Table 3.70. It can be seen that:

- Increasing the learning rate from 0.005 to 0.1 resulted in better 14 day forecasts. This is in agreement with the results obtained in Section 3.5.3.3.
- The network with one hidden layer performed better than the network with two hidden layers when a learning rate of 0.005 was used (models SMB_M_A_66_10_88 and SMB_M_A_63_10_88). However, the opposite was true when a learning rate of 0.1 was used, suggesting that factors other than network geometry (e.g. testing frequency) might be responsible for most of the small variations in the RMS prediction errors obtained when using networks with various geometries in Section 3.5.3.4. This result was further confirmed when model SMB_M_A_71_10_88 was re-trained, and a testing frequency of 16,710 was used instead of a testing frequency of 5,000. The best results obtained (Table 3.71) show that despite the fact that the same network parameters and geometries were used, there was quite a significant difference in the best 14 day forecasts obtained as a result of the different testing frequencies used. This reinforces the theory put forward in Section 3.5.3.3 that networks should be tested more frequently when larger learning rates are used.

Table 3.70: Comparison of the Best Results Obtained for Models with One and Two Hidden Layers and Learning Rates of 0.005 and 0.1

Model	N^{H1} - N^{H2}	Learning rate (η)	RMSE (EC units)
SMB_M_A_66_10_88	30-0	0.005	40.9
SMB_M_A_70_10_88	30-0	0.1	40.4
SMB_M_A_63_10_88	60-20	0.005	41.9
SMB_M_A_65_10_88	60-20	0.1	39.0

Table 3.71: Comparison of Best Results Obtained for Model SMB_M_A_71_10_88 when Using Different Testing Frequencies

Testing frequency	RMSE (EC units)
5,000	39.0
16,710	40.6

By using a fixed learning rate of 0.1 and a fixed momentum value of 0.6, instead of the default values (Table 3.39), and by increasing the number of nodes in the first hidden layer from 45 to 60 and the number of nodes in the second hidden layer from 15 to 20, the RMSE of the best 14 day forecast for 1988 was able to be reduced from 44.8 EC units (model SMB_M_A_18_10_88) to 39.0 EC units (model SMB_M_A_65_10_88).

Models were also developed to obtain 14 day salinity forecasts for 1989 (model SMB_M_A_65_10_89), 1990 (model SMB_M_A_65_10_90) and 1991 (model SMB_M_A_65_10_89) using the geometry and internal parameters that were found to result in the optimum forecasts for 1988 (model SMB_M_A_65_10_88).

A comparison of the best results obtained using the default settings (models SMB_M_A_18_10_88, SMB_M_A_18_10_89, SMB_M_A_18_10_90 and SMB_M_A_18_10_91) and the "optimum" settings (models SMB_M_A_65_10_88, SMB_M_A_65_10_89, SMB_M_A_65_10_90 and SMB_M_A_65_10_91) for the various years is shown in Table 3.72. It can be seen that:

- Significant improvements in the forecasts for 1988 and 1989 were obtained by using the "optimum" parameters.
- The forecasts for 1990 and 1991 were noticeably better when the default settings were used.
- On average, the forecasts obtained using the "optimum" parameters were slightly better than those obtained using the default settings.

Table 3.72: Best Results Obtained Using Default and "Optimum" Settings

Year predicted	RMSE (EC units)	
	Default settings	"Optimum" settings
1988	44.8	39.0
1989	49.3	44.3
1990	44.5	48.4
1991	45.8	49.4
Average	46.1	45.3

The above results indicate that the optimality of the settings is dependent on the data set used, as this defines the error surface in weight space. However, the testing frequency also has a major influence on the results obtained (as discussed in Section 3.5.3.3), making it difficult to draw a direct comparison between results obtained for networks using different internal parameters and geometries.

3.5.3.6 Generalisation ability at various stages of learning

The results obtained in Sections 3.5.3.3 and 3.5.3.4 indicate that in most cases, the generalisation ability of the models increases gradually as learning progresses. This is indicated by a constant decrease in the average RMS prediction error, until a local minimum in the error surface is reached. In this section, the various stages of learning, and the behaviour of the networks as they approach a local minimum in the error surface, are examined in detail, with the aim of devising some guidelines for optimising the generalisation ability of neural network models.

Stages of learning

In order to examine the various stages of learning in detail, the size of the steps taken in weight space was chosen to be very small by using a learning rate of 0.001 in conjunction with a momentum value of 0.0 (model SMB_M_A_72_10_88). The RMSEs at the various stages of learning are shown in Figures 3.110 and 3.111 and in Appendix E. The RMSE of the best 14 day forecast and the learn count at which it was obtained are shown in Table 3.73. It can be seen that:

- Learning progressed very slowly, and the best result was obtained at a learn count of 2,400,000.
- Once the vicinity of the local minimum in the error surface was reached, oscillations in the RMSE still occurred, but the magnitude of the oscillations was very small.
- The best RMS prediction error obtained was higher than those obtained when larger learning rates were used (Section 3.5.3.3), thus confirming the results obtained earlier.

Table 3.73: Best Result Obtained Using Model SMB_M_A_72_10_88

Model	Learning rate (η)	RMSE (EC units)	Learn count
SMB_M_A_72_10_88	0.001	43.0	2,400,000

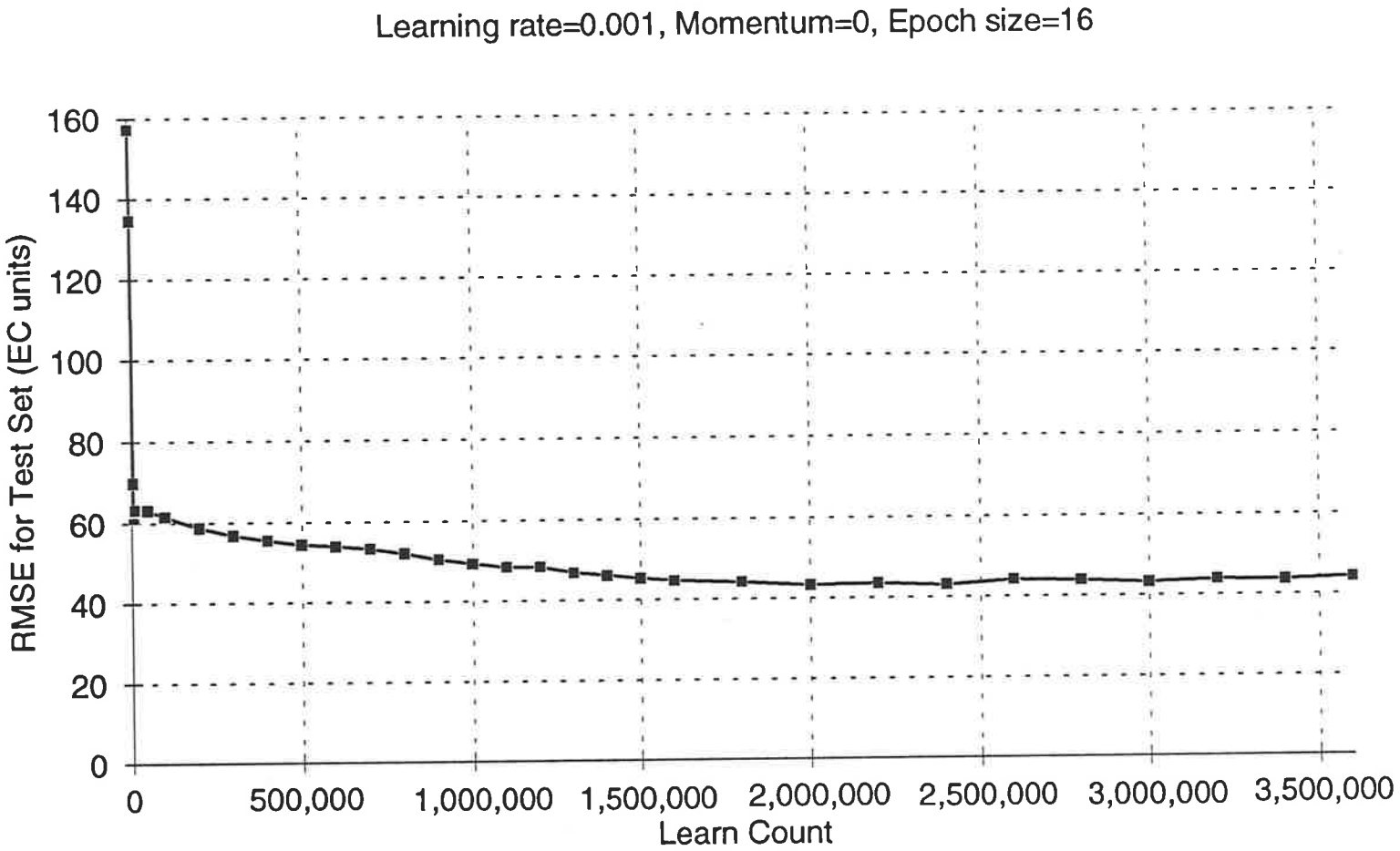


Figure 3.110: RMSE at Different Stages of Learning for Model SMB_M_A_72_10_88

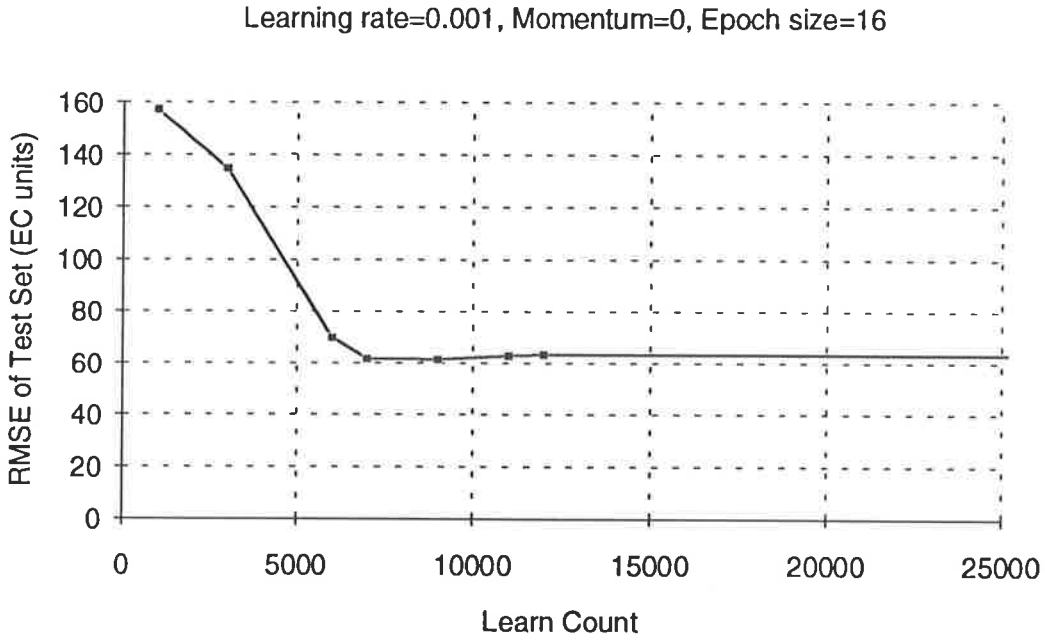


Figure 3.111: RMSE at Various Stages of Learning for Model SMB_M_A_72_10_88 (Up to a Learn Count of 25,000)

When testing model SMB_M_A_72_10_88, the three stages of learning described by Cheung et al. (1990), namely the error-convergent stage, the competition stage and the domination stage (see Section 2.1.8.1), were clearly visible, as learning progressed very slowly. In model SMB_M_A_72_10_88, the error-convergent stage was very short, which is in agreement with the observations made by Cheung et al. (1990). It was represented by the first 35 iterations, during which the RMS prediction error of the **normalised training set** (see Section 3.3.4.1) converged to a mean value of 0.38 (not shown here).

The competition stage was also short, indicating that some dominant patterns exist. The competition stage extended to a learn count of approximately 2,700, during which time the RMS prediction error of the **normalised training set** (not shown here) remained fairly stable in the vicinity of 0.38. This is short compared with the duration of the entire learning process. The plot comparing predicted with actual salinities for the test set at a learn count of 1,000 (Figure 3.112) clearly shows that no patterns had been dominantly trained at this stage.

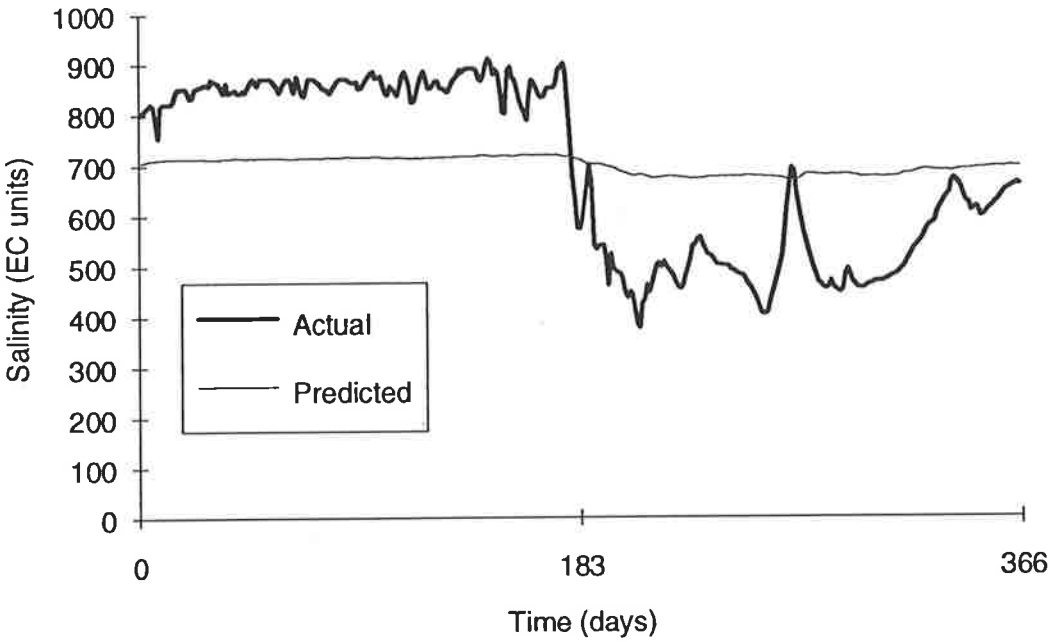


Figure 3.112: Actual and Predicted Salinities at a Learn Count of 1,000 - Model SMB_M_A_72_10_88

After a learn count of approximately 2,700, the domination stage commenced, which was characterised by a constant drop in the RMSE of the **normalised training data** (not shown here). As shown in Figures 3.113 to 3.115, the dominantly trained pattern was that of low-flow salinities. This is not surprising, as there are many more examples of the low-flow salinity pattern in the training set than any others. With the epoch size set at 16, 16 examples are presented to the network before the weights are updated. Of the 16 examples presented to the network, the majority are examples of the low-flow salinity pattern. As a result, the majority of the errors are due to the low-flow salinity pattern, and, as the weights are changed in a manner that reduces the error by the largest possible amount, the errors due to the low-flow salinity pattern are reduced more than those of the other patterns. Consequently, this pattern is dominantly trained.

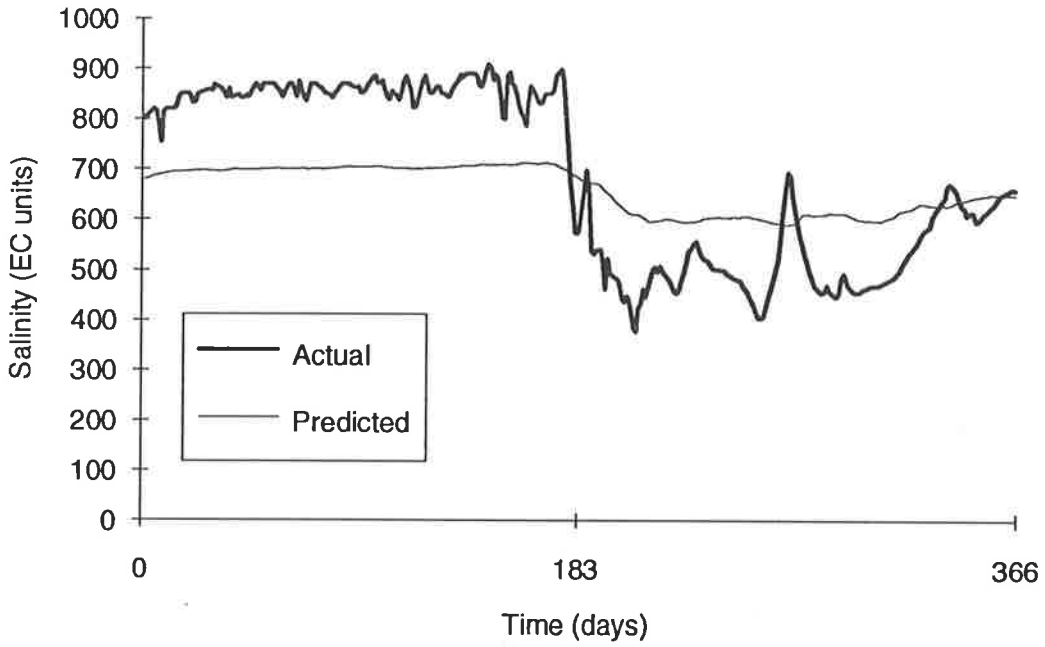


Figure 3.113: Actual and Predicted Salinities at a Learn Count of 3,000 - Model SMB_M_A_72_10_88

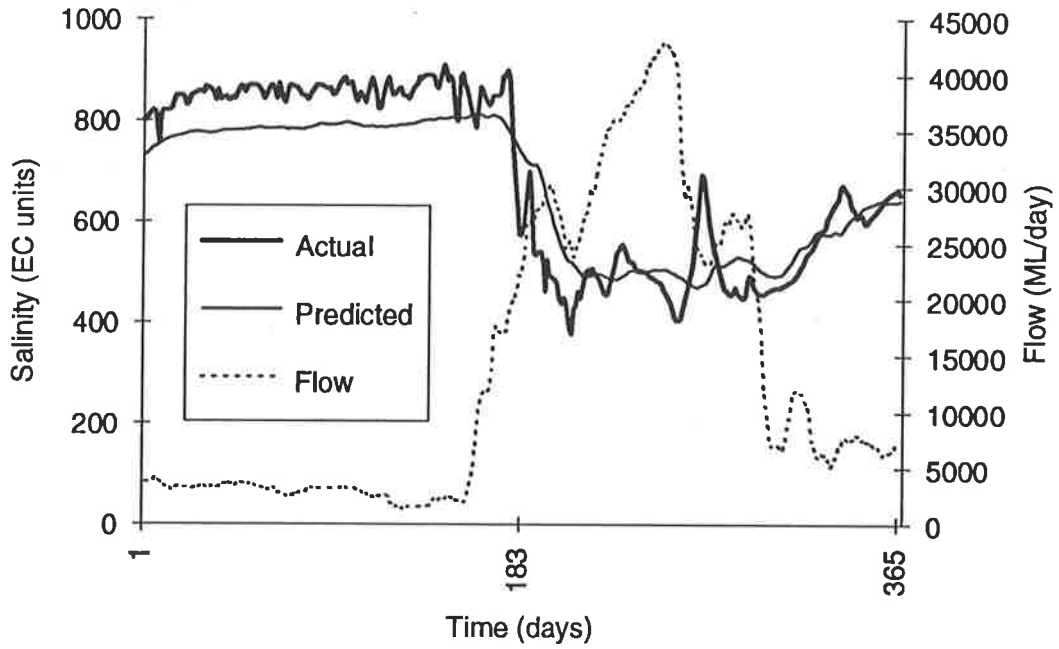


Figure 3.114: Actual and Predicted Salinities at a Learn Count of 6,000 - Model SMB_M_A_72_10_88

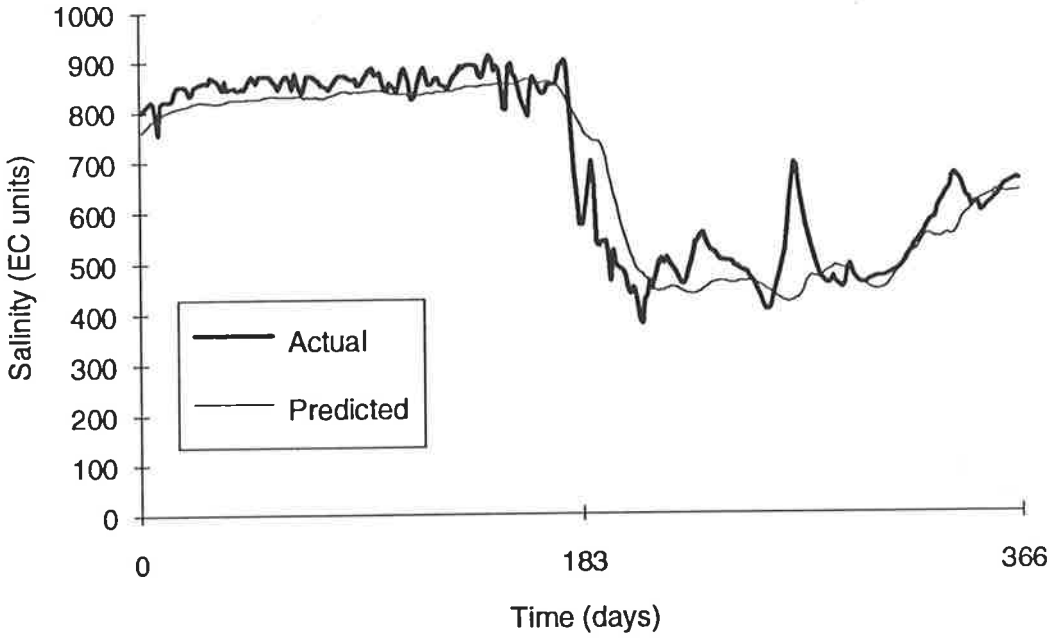


Figure 3.115: Actual and Predicted Salinities at a Learn Count of 9,000 - Model SMB_M_A_72_10_88

The rapid manner in which the low-flow salinity pattern is learned is also illustrated in Figures 3.110 and 3.111, which shows a drop in the RMSE of the test set from 157.2 EC units to 61.6 EC units over the presentation of 6,000 training examples. The domination situation begins to disappear at a learn count of approximately 9,000. At this point, the errors produced by the low-flow salinity pattern are sufficiently small for the errors of the less frequent patterns, although fewer in number, to dominate the overall error. Consequently, the less frequent patterns are learnt as shown in Figures 3.116 and 3.117.

Figure 3.110 indicates that the less frequent patterns are being learnt more slowly than the low-flow salinity pattern. This can be attributed to the fact that the number of patterns to be learned is greater, as well as the fact that examples of these relationships occur less frequently in the training samples and hence take longer to learn. As the less frequent patterns are being learned, the RMSE of the test set decreased from 61.6 EC units to 43.0 EC units over the presentation of 2,393,000 training examples (Figure 3.110). During the same period, the RMSE of the **normalised training set** (not shown here) stabilised near 0.06.

In many instances, neural networks dominantly train frequent patterns, while neglecting to capture the relationships of the infrequent patterns, resulting in poor local minima and poorly trained patterns (Cheung et al., 1990). This phenomenon does not seem to

occur with the tests carried out in this research. The domination situation disappears and the relationship of the less frequent patterns is captured.

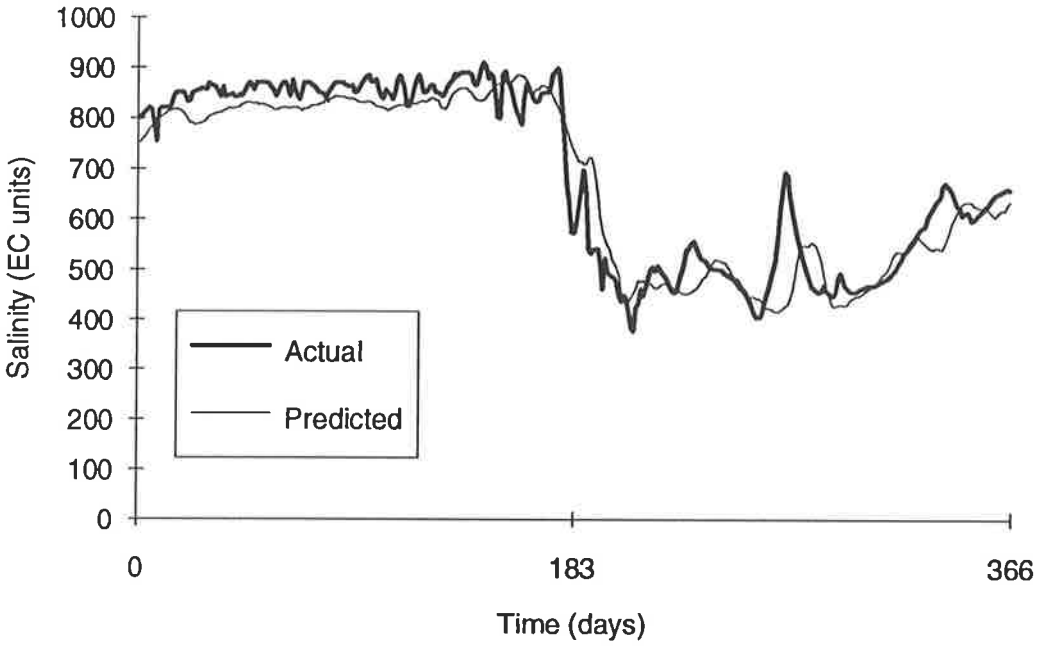


Figure 3.116: Actual and Predicted Salinities at a Learn Count of 320,000 - Model SMB_M_A_72_10_88

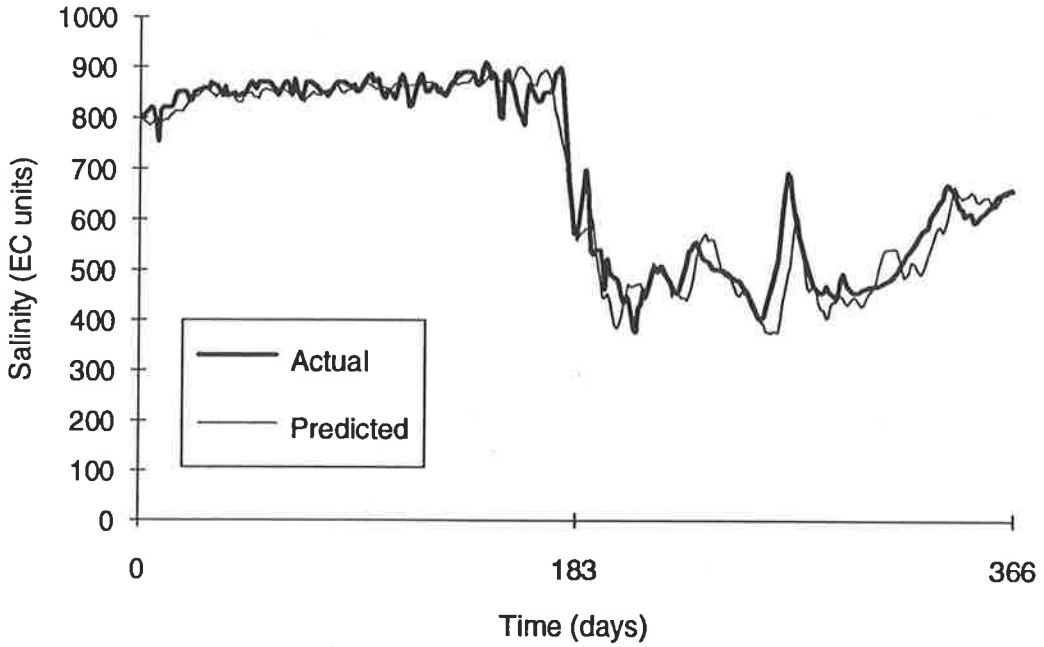


Figure 3.117: Actual and Predicted Salinities at a Learn Count of 3,400,000 - Model SMB_M_A_72_10_88

Network behaviour in the vicinity of local minima

When small steps are taken in weight space (e.g. model SMB_M_A_72_10_88), the oscillations in the RMS prediction error at the various stages of learning are minimal, and consequently, once the vicinity of a local minimum in the error surface is reached, there is no need for frequent testing. Consequently, the result obtained is virtually independent of when training is stopped and the network is tested. However, when small steps are taken in weight space, the time taken for training is excessively long (e.g. approximately 28 hours for model SMB_M_A_72_10_88), and the network does not have the ability to escape local minima in the error surface.

As discussed in Section 3.5.3.3, by taking larger steps in weight space, training time can be reduced dramatically (e.g. to approximately 30 minutes for model SMB_M_A_44_10_88), and the ability to find "better" local minima in the error surface is increased. This is especially true for large networks, as there are a large number of connection weights, and consequently many local minima exist in the error surface. However, as discussed in Section 3.5.3.3, when larger steps are taken in weight space, there is an increase in the magnitude of the oscillations in the RMS prediction error that occur when a local minimum in the error surface is reached. Consequently, the generalisation ability of the network is dependent on when training is stopped (i.e. at a peak or a trough in the RMSE).

In order to investigate methods for optimising the generalisation ability of ANN models, while maintaining high training speeds, it is important to have a good understanding of the causes and the frequency of the oscillations in the RMS prediction error that occur once a local minimum in the error surface is reached.

According to the theory outlined in Section 3.5.3.3, once a network is in the vicinity of a local minimum, oscillations in the RMSE should occur with every weight update (unless the steps taken in weight space are extremely small). For most of the tests carried out in Sections 3.5.3.3 and 3.5.3.4, the epoch size used was 16, so that once a network approaches a local minimum in the error surface, oscillations in the RMS prediction error should occur after the presentation of every 16 training samples. However, for most models, testing was carried out at intervals of 5,000 or 10,000 training samples. It was decided to adopt a testing interval (frequency) of 16, in order to investigate the effect that each weight update has on the RMS prediction error.

Model SMB_M_A_19_11_88 (Section 3.5.2.5) was used for this purpose. When developing model SMB_M_A_19_11_88, a testing interval of 5,000 was used. The

results obtained (Appendix E) indicate that the vicinity of a local minimum was reached at a learn count of 15,000, at which point continued training resulted in oscillations in the RMSE. At a learn count of 15,000, the RMSE was 44.5 EC units. The lowest RMSE of 41.0 EC units was obtained at a learn count of 35,000. Training was concluded at a learn count of 55,000, at which point the RMSE was 41.5 EC units.

Training for model SMB_M_A_19_11_88 was re-commenced at learn counts of 15,000, 35,000 and 55,000, and the models tested at intervals of 16 training samples. The results in Figure 3.118 show that:

- Oscillations in RMSE occur after each weight update, as suggested by the theory outlined in Section 3.5.3.3.
- The magnitude of the oscillations is greater at a learn count of 15,000+, as the learning rate and momentum, and hence the size of the steps taken in weight space, were greater than those at learn counts of 35,000+ and 55,000+. It should be noted that in model SMB_M_A_19_11_88, the learning rate and momentum are reduced at a learn count of 30,000, as the default learning rates and momentums are used (Table 3.39).
- Rises and falls in the RMSE occurred at the same time, regardless of the learn count at which training was re-commenced. Consequently, it can be concluded that the changes in RMSE are a function of the training samples that are presented to the network between weight updates (it should be noted that the order in which the training samples were presented to the networks was the same in each case). The reason for this is that the size of the changes made to the connection weights is proportional to the error between actual and predicted values obtained during training. If the 16 training samples presented to the network between weight updates have been learnt well, the errors between the actual and predicted values will be small, and hence a smaller step will be taken in weight space. However, if the 16 training samples presented to the network between weight updates contain one or more patterns that have not been learnt well, the errors between the actual and predicted values will be larger, and hence bigger steps will be taken in weight space.
- An RMSE of 40.70 EC units was obtained at a learn count of 35,456, which is slightly better than the previous best RMSE of 41.0 EC units, which was obtained at a learn count of 35,000.

The above results indicate that once a network approaches a local minimum in the error surface:

- Oscillations in the RMS prediction error occur after each weight update.

- The magnitude of the oscillations is a function of learning rate, momentum and the training samples presented to the network between weight updates.

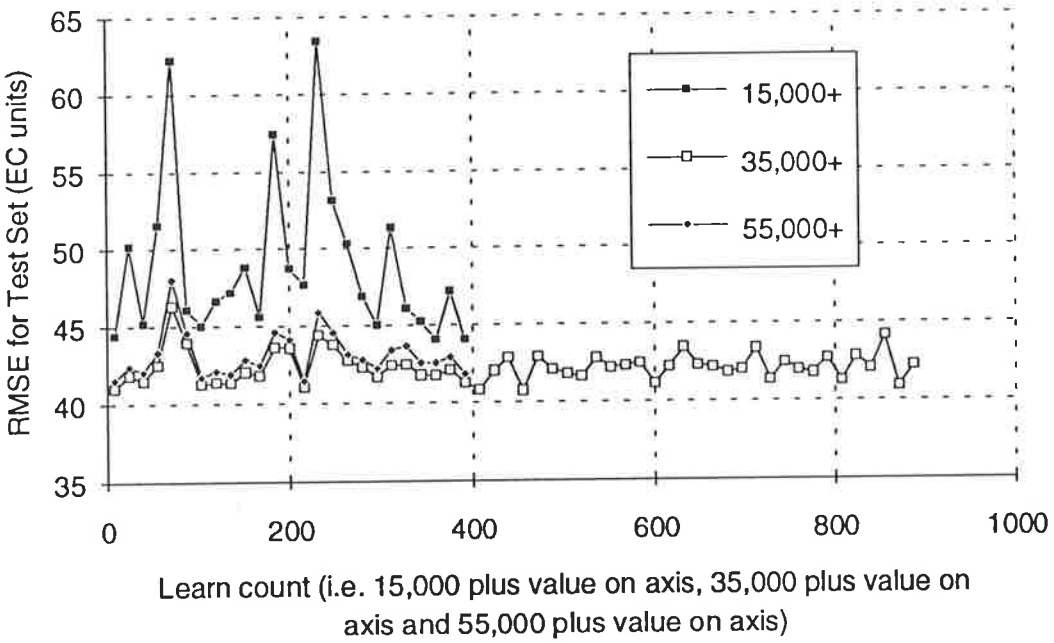


Figure 3.118: Changes in RMS Prediction Error After Each Weight Update for Model SMB_M_A_19_11_88

It is also interesting to inspect the plots of the forecasts obtained from one weight update to the next. The changes in RMS prediction error appear to be caused by whole portions of the graph moving "up" or "down", while the shape of the plot is retained (i.e. the outputs are increased or decreased by the same amount). This phenomenon is clearly illustrated in Figure 3.119, and is typical for all models. The plot shows the actual salinities as well as the predicted salinities at a learn count of N and at a learn count of $N + \varepsilon$.

This phenomenon might be explained by examining the weight update equation for the connections between the second hidden layer nodes and the output nodes. The update equation for the connection weight joining node i in the second hidden layer to node j in the output layer is given in Equation 3.3, and is obtained by combining Equations 2.16 and 2.18.

$$\Delta w_{ji} = \eta \, (o_j^d - o_j^p) \, F'(I_j) \, x_i \qquad (3.3)$$

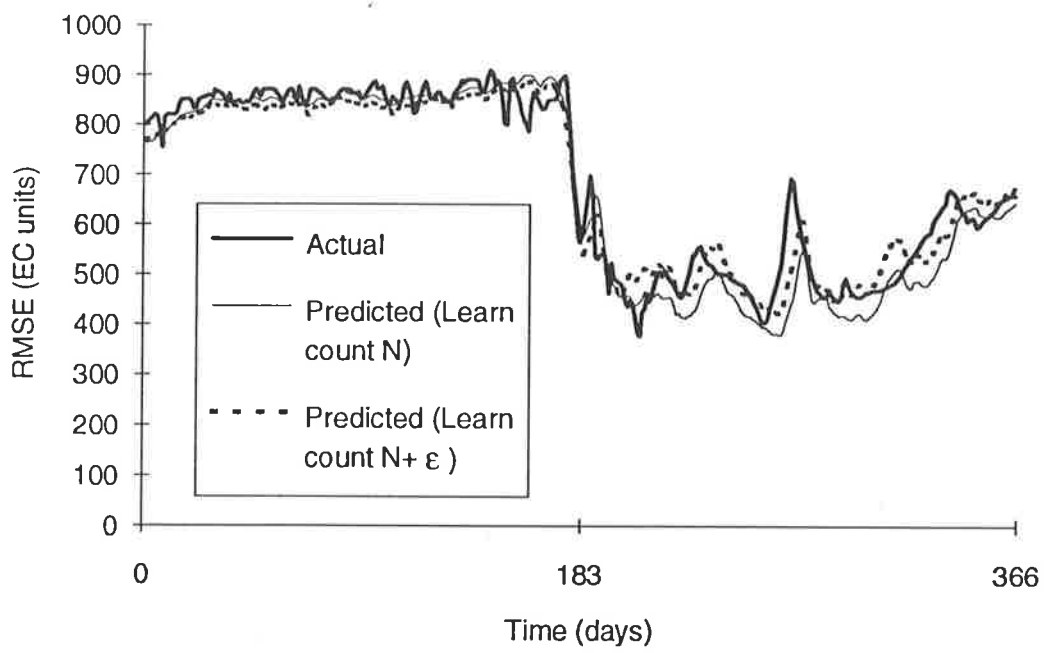


Figure 3.119: Typical Plot Showing the Phenomenon of Whole Portions of the Plot of Predicted Salinities Moving Up or Down from One Weight Update to the Next

For a particular weight update, the error and the learning rate are the same for all connections between node j in the output layer and the nodes in the second hidden layer. Consequently, Equation 3.3 may be re-written as follows:

$$\Delta w_{ji} = a_1 F'(I_j) x_i \qquad a_1 = \text{constant} \tag{3.4}$$

At the start of training, the output of each node tends to be small. Consequently, the derivative of the transfer function, with respect to the input, as well as the changes in the connection weights, are large (Maren et al., 1990). As learning progresses, the node outputs approach one of two stable states (+1 or -1), and consequently the derivative of the transfer function with respect to the input, as well as the changes in the connection weights, are small (Maren et al., 1990). Therefore, once the network approaches a local minimum in the error surface, one can assume that all the patterns have been learnt and that all node outputs approach either +1 or -1. As a result, the derivative of the transfer function, with respect to the input, may be considered to be constant for each node. Equation 3.4 may thus be re-written as follows

$$\Delta w_{ji} = a_2 x_i \qquad a_2 = \text{constant} \tag{3.5}$$

However, as discussed above, once all the patterns have been learned, the node activations (i.e. x_i) can be assumed to approach either +1 or -1. As a result, the weight

change for each node is equal to $+a_2$ or $-a_2$. This means that each time a weight update occurs, all outputs are changed by a value of $+a_2$ or $-a_2$. The variations in the magnitude of a_2 from one weight update to the next are generally due to different magnitudes in the error term (Equation 3.3).

It should be noted that in practice, there can be small variations in a_2 (i.e. a_2 is not exactly the same for all connections between the nodes in the second hidden layer and the output node), as the assumption that all node outputs are +1 or -1 does not generally hold. However, any variations in a_2 can be expected to be small.

It can be seen from Figure 3.119 that once the weights have been updated (i.e. at a learn count of $N + \varepsilon$), the values of the low flow salinities at the beginning of the year have decreased by a fixed amount and that the values of the high flow salinities in the second half of the year have increased by a fixed amount. It should be noted that the magnitude of the decrease in the low flow salinity pattern is less than the magnitude of the increase in the high flow salinity patterns. This suggests that the low flow salinity patterns have been learnt better than the high flow salinity patterns. It is likely that changes in the sign of the error term (Equation 3.3) from one weight update to the next are responsible for whole portions of the graph moving "up" and "down" as training progresses.

As discussed above, the magnitude of the oscillations in RMS prediction error is a function of which training samples are presented to the network between weight updates. Large variations in the error are caused by the presence of poorly trained patterns in any particular epoch. Cheung et al. (1990) suggest that poorly trained patterns can be eliminated by increasing the proportion of such patterns in the training set, thus ensuring an equal representation of all patterns. Consequently, each pattern should have approximately equal representation within one epoch, eliminating the dependency of the RMS prediction error on which patterns make up the epoch presented to the network between weight updates.

In the case study considered, high flow salinities may be considered poorly trained patterns, as discussed earlier in this section. Training / testing set 10 was "spiked" to increase the proportion of the high flow salinity patterns. In training / testing set 10a, the representation of the high flow salinity patterns was increased by doubling the number of samples during high flow conditions and reducing the number of samples during low flow conditions. Consequently, the total number of samples in the training set remained approximately constant (it was reduced from 1,671 to 1,602). In training / testing set 10b, the representation of the high flow salinity pattern was increased by quadrupling the number of examples during high flow conditions while leaving the

number of examples during low flow conditions unchanged. Consequently, the total number of samples in the training set was increased from 1,671 to 4,057.

Model SMB_M_A_20_10_88 was re-trained using training / testing sets 10a (model SMB_M_A_20_10a_88) and 10b (model SMB_M_A_20_10b_88). Details of the network parameters and geometries are summarised in Appendix D. The results in Table 3.74 and Figure 3.120 show that "spiking" the training / testing set did not improve network performance. The best predictions obtained using the "spiked" training / testing sets (10a and 10b) were very similar. However, they were worse than the best forecast obtained using training / testing set 10.

Table 3.74: Best Results Obtained When Using "Spiked" Training / Testing Sets (ε = 5)

Model	RMSE (EC units)	Learn count
SMB_M_A_20_10_88	42.0	105,000
SMB_M_A_20_10a_88	46.8	70,000
SMB_M_A_20_10b_88	45.7	80,000

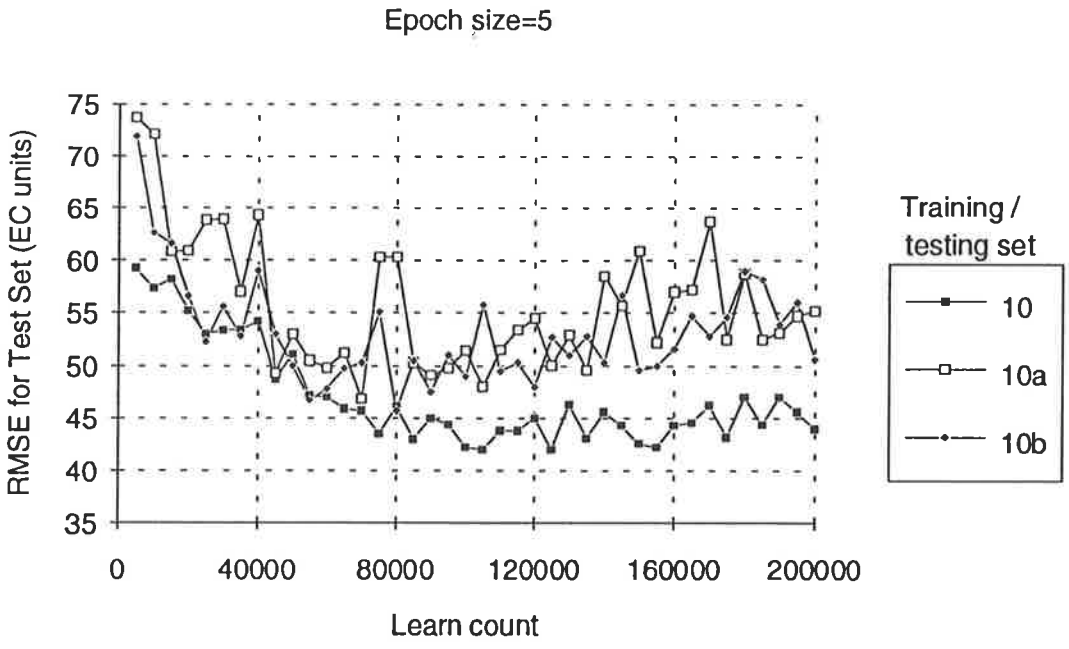


Figure 3.120: RMSE at Different Stages of Learning for Models With "Spiked" Training / Testing Sets (ε = 5)

Model SMB_M_A_26_10_88, which uses an epoch size of 365 rather than an epoch size of 5, was also repeated using training sets 10a (model SMB_M_A_26_10a_88) and 10b (model SMB_M_A_26_10a_88). Details of the network parameters and

geometries are summarised in Appendix D. The results shown in Table 3.75 and Figure 3.121 confirm the results obtained when using models SMB_M_A_20_10_88, SMB_M_A_20_10a_88 and SMB_M_A_20_10b_88.

Table 3.75: Best Results Obtained When Using "Spiked" Training / Testing Sets ($\varepsilon = 365$)

Model	RMSE (EC units)	Learn count
SMB_M_A_26_10_88	42.5	800,000
SMB_M_A_26_10a_88	47.1	840,000
SMB_M_A_26_10b_88	47.2	680,000

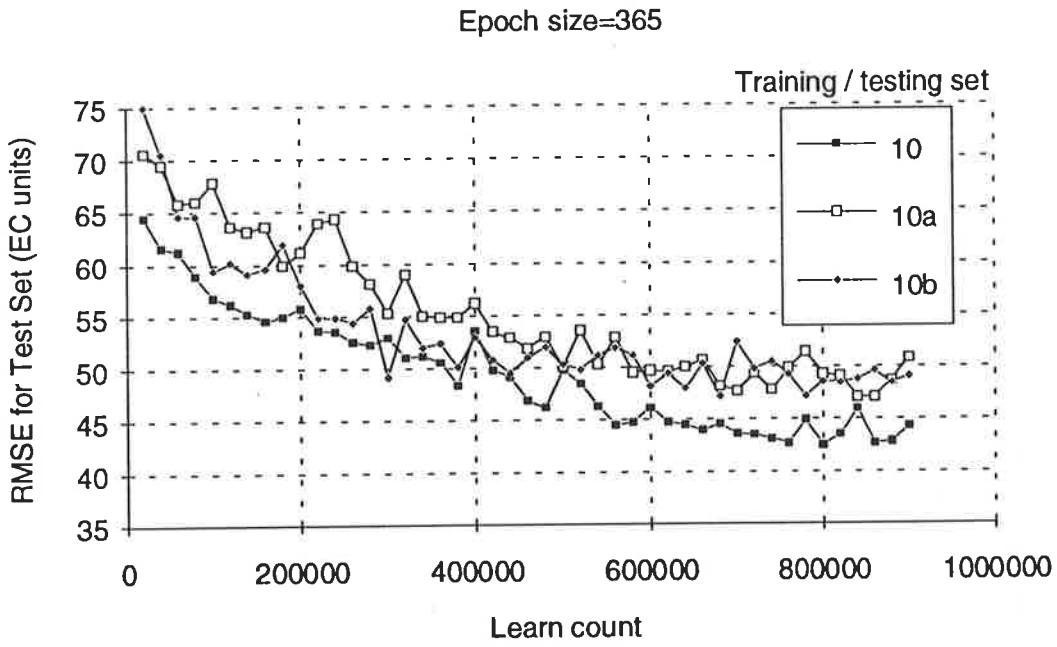


Figure 3.121: RMSE at Different Stages of Learning for Models With "Spiked" Training / Testing Sets ($\varepsilon = 365$)

A method for optimising generalisation ability

It has been suggested (NeuralWare, Inc., 1991) that learning rates and momentum values should be reduced as learning progresses. This increases learning speed in the initial stages of learning, as the steps taken in weight space are large. As training continues, and the network approaches a local minimum in the error surface, the size of the steps taken in weight space is reduced, decreasing the magnitude of the oscillations in the RMSE. However, the application of this theory presents a number of practical problems including:

- It is difficult to know when, and by how much, to reduce the learning rates and momentum values. If the step sizes are reduced while the network is still on a shallow portion of the error surface, learning speed is reduced dramatically. In addition, networks are unable to escape local minima in the error surface. The step sizes may also be reduced too late. If the network reaches a local minimum while the step sizes taken are still large, divergent behaviour may occur, and the network may be in an undesirable region of the weight space when the step sizes are reduced.
- The method is perceived to work in a manner similar to simulated annealing. However, this is not the case. In simulated annealing, as the sizes of the steps taken in weight space are gradually reduced, the network is able to escape the shallower local minima in the error surface, but not steeper ones. However, this is not the case here, as the network does not have to jump "over" the peaks in the error surface as illustrated in Figure 3.122. Even though the network is in a deep trough at point A, only a small weight change is required to move the network to a different position in weight space (out of the deep trough). In the case of simulated annealing, a lot of energy would be required for the network to "jump out" of the trough. Consequently, gradually reducing the step sizes does not ensure that once the smaller step sizes are used, the network is in one of the deeper troughs in the error surface. This is particularly true when training large networks, as many local minima exist.

A more satisfactory method for obtaining the best possible forecasts is given below. The best and quickest way of finding a "deep" local minimum in the error surface is to use relatively large step sizes in weight space and to carry out testing at regular intervals. Once oscillations in the RMS prediction error start to occur, networks should be tested more frequently in order to increase the chances of obtaining a better result. The network producing the best result from these "spot checks" should be retained. However, at this point, the network is not necessarily at a local minimum in the error surface. Training should be continued with a very small learning rate and momentum and an epoch size equal to the size of the training set, in an attempt to reach the bottom of the closest local minimum. By setting the epoch size equal to the number of examples in the training set, the weight updates are such that the error is reduced over the entire training set, and not over a subset, which might contain some poorly trained patterns, resulting in large errors and large steps in weight space. The only time when using an epoch size equal to number of examples in the training set results in an increase in the RMS prediction error is when one of the following holds:

- The steps taken in weight space are too large (e.g. the learning rate and/or momentum are too large).
- Overtraining occurs.

- The training set is not representative of the test set.
- A local minimum in the error surface has been reached.

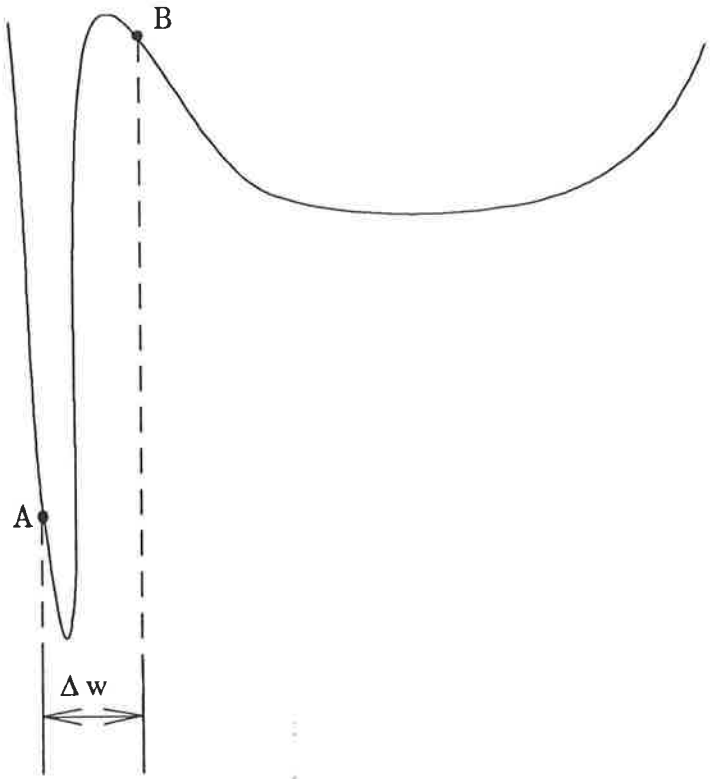


Figure 3.122: Example of an Error Surface that Enables Local Minima to be Escaped when Small Steps are Taken in Weight Space

The above method was applied to model SMB_M_A_18_10_88. It should be noted that the test had to be repeated with a testing frequency of 1671 (the number of samples in the training set), so that once the epoch size was increased from 16 to 1671, the whole training set was presented to the network before the next weight update was carried out. The reason for this is that the software package used to simulate neural network operation (NeuralWorks Explorer) updates the connection weights once the total learn count is equal to a multiple of the epoch size. By using a testing frequency of 1671, the best result is obtained at a learn count equal to a multiple of 1671. This ensures that after the epoch size is changed to 1671, and training is continued, 1671 samples are presented to the network before the next weight update is carried out.

The best result obtained when testing at intervals of 1671 was 41.68 EC units at a learn count of 23,394. At that point, the learning rate and the momentum were divided by 100 and the epoch size was increased from 16 to 1671. Training was continued and the network tested after every weight update. The results obtained are summarised in Table 3.76. It can be seen that after the first weight update, the RMS prediction error

decreased from 41.68 EC units to 41.46 EC units. Continued training had no effect on RMSE, suggesting that a local minimum in the error surface had been reached.

Table 3.76: Result Obtained When Applying the Procedure for Reaching a Local Minimum in the Error Surface to model SMB_M_A_18_10_88

Learn count	RMSE (EC units)
23,395	41.68
25,066	41.46
26,737	41.46
28,408	41.46
30,079	41.46
31,750	41.46

Nature of error surface in the vicinity of local minima

The nature of the error surface in the vicinity of the local minimum obtained at a learn count of 25,066 for model SMB_M_A_18_10_88 (Table 3.76) was explored, in order to determine whether many local minima exist in the error surface, which is what one would expect for the size of network used. This was done by carrying out the following procedure:

1. The weights obtained for model SMB_M_A_18_10_88 at a learn count of 25,066 were "jogged" with the aid of NeuralWorks Professional II/Plus. "Jogging" adds a random value to each of the connection weights. The upper and lower limits of the values to be added or subtracted are specified (the "jogging range") by the user. The "jogging ranges" used include ±0.002 and ±0.01.

2. The test set was presented to the network with the "jogged" connection weights, and the RMSEs between the actual and predicted (14 day forecast) values were calculated (Table 3.77).

3. Training was continued and the network tested after each weight update.

Starting with the connection weights obtained for model SMB_M_A_18_10_88 at a learn count of 25,066 on each occasion, the weights were "jogged" six separate times using a jogging range of ±0.002 and six separate times using a jogging range of ±0.01 (Table 3.77).

Table 3.77: RMS Prediction Errors (EC units) After Jogging

Jogging Range	
±0.002	±0.01
41.70	43.73
41.72	41.40*
41.28*	42.68
41.51*	41.98
41.38*	41.60*
41.54*	44.58

The results in Table 3.77 indicate that:

- When a "jogging range" of ±0.002 was used, there was very little change in network performance after jogging. There were a number of results that were better than the starting value of 41.68 EC units, as indicated by the asterisk (*) following the number. The best result obtained was 41.28 EC units. All of the values obtained after jogging constituted a local minimum in the error surface (as indicated by the shaded background), as further training did not result in any changes in the RMS forecasting error.
- When a "jogging range" of ±0.01 was used, there were some significant changes in network performance. The results obtained varied between 41.40 EC units and 44.58 EC units. Two of the results were better than the starting value of 41.68 EC units, as indicated by the asterisk (*). Three of the six values constituted local minima in the error surface, as indicated by the shaded background. The other three results were significantly greater than the starting value of 41.68 EC units, and, as one would expect, did not constitute local minima in the error surface as indicated by a steady (although small) decrease in the RMS prediction error with continued training.

The results obtained demonstrate that for the network used, there exist many local minima that are very close together. The results also suggest that the error surface is similar to that shown in Figure 3.123 (two-dimensional simplification), with a minimum "plateau", that has many smaller peaks and troughs at its base.

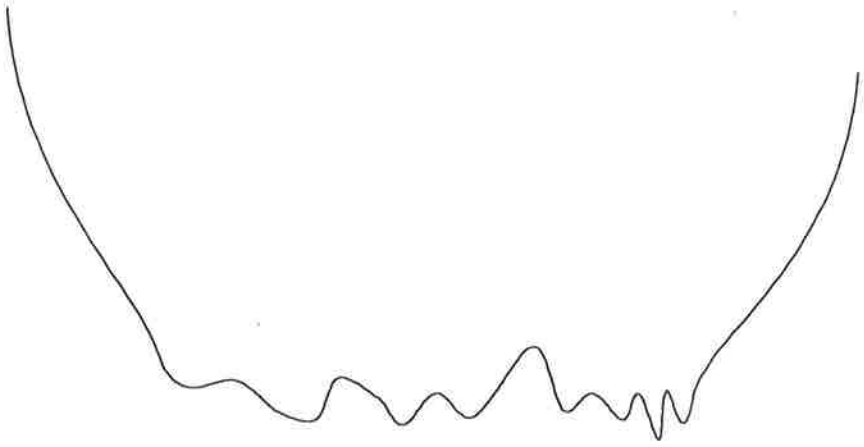


Figure 3.123: Typical Minimum "Plateau" in the Error Surface for Large Networks

Implications for real time forecasting applications

The results obtained in Sections 3.5.3.3 and 3.5.3.4 indicate that as training progresses, there is a steady decrease in the average RMS forecasting error (FE_{ave}) (Figure 3.124, region AB). With continued training, a local minimum in the error surface is reached (point B), and the average RMS forecasting error remains constant as training progresses ($FE_{plat,ave}$) (Figure 3.124, region BC). However, although the average RMS forecasting error remains constant with time, the RMS forecasting errors at different learn counts vary between $FE_{plat,min}$ and $FE_{plat,max}$. The magnitude of the oscillations in the RMS forecasting error (i.e. $FE_{plat,max}$ - $FE_{plat,min}$) is a function of the size of the steps taken in weight space.

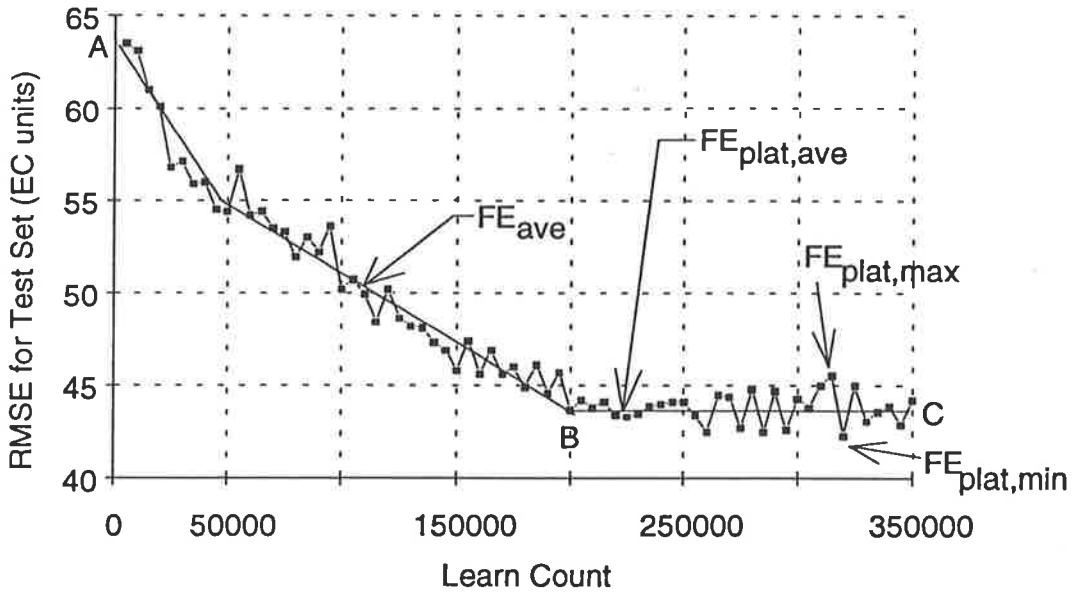


Figure 3.124: Typical Generalisation Ability of a Neural Network Model During Training

The aim of training is to find a set of connection weights that will minimise the RMS forecasting error in the shortest possible training time. In order to maximise the network's generalisation ability, training should be continued until a local minium in the error surface has been reached (i.e until region BC has been reached, Figure 3.124). Once a local minimum in the error surface has been reached, training should be stopped when the RMS forecasting error is close to, or preferably at, $FE_{plat,min}$. In order to maximise training speed, larger steps should be taken in weight space. By using larger steps in weight space, the networks are also able to escape local minima in the error surface, and are thus capable of achieving better generalisation ability (i.e. lower values of $FE_{plat,min}$). However, by using larger steps in weight space, the magnitude of the oscillations in the RMS forecasting error (i.e. $FE_{plat,max}$ - $FE_{plat,min}$) is also greater, and if the steps taken in weight space are too large, divergent behaviour may occur.

The network behaviour, and aims of training, described above have a number of implications for users of back-propagation networks, especially for real time forecasting applications. If cross-validation is to be used as a stopping criterion, three data sets have to be used; a training set, a validation set and a forecasting set. The procedures outlined in Section 3.5.3.6 can be used to obtain the best possible forecast for the validation set. The network that gives the best forecast when the validation set is used is assumed to give the best forecast when the forecasting set is used. However, for this to be the case, it is crucial that the validation set is representative of the general relationship to be approximated.

In many real life applications, the available data are limited, in which case it might not be feasible to have a validation set that is separate from the forecasting set, because as much information as possible is required for training purposes. However, if the real time forecasting ability of the model is to be tested, the forecasting set must not be used in the development of the model, and cross-validation cannot be used as a stopping criterion. In such situations, it may be more appropriate to present a fixed number of training samples to the network during training.

When a fixed number of training samples are presented to the network, it is vital to know when the local minimum in the error surface will be reached for a given set of parameters. If this is not the case, training might be stopped too early (i.e. when the RMS forecasting error is still decreasing steadily - region AB, Figure 3.124), or overtraining might occur. The magnitude of the oscillations in the RMS forecasting error, once a local minimum in the error surface has been reached, also has to be known. When training is stopped in region BC (Figure 3.124), the RMS forecasting error (FE)

could be anywhere between $FE_{plat,min}$ and $FE_{plat,max}$ (i.e. $FE_{plat,min} \leq FE \leq FE_{plat,max}$) for a given network, depending on when training was stopped. Consequently, in order to maximise the generalisation ability of the model, a compromise has to be made with regards to the size of the steps taken in weight space. As discussed above, the minimum RMS forecasting error that is obtained when relatively large steps are taken in weight space ($(FE_{plat,min})_{large}$) is generally smaller than that obtained when relatively small steps are taken in weight space ($(FE_{plat,min})_{small}$) (Figure 3.125). However, the magnitude of the oscillations in the RMS forecasting error are bigger when larger steps are taken in weight space (i.e. $(FE_{plat,max} - FE_{plat,min})_{large} > (FE_{plat,max} - FE_{plat,min})_{small}$). As a result, despite the fact that $(FE_{plat,min})_{large} < (FE_{plat,min})_{small}$, $(FE_{plat,max})_{large}$ may still be bigger than $(FE_{plat,max})_{small}$, which is undesirable. In other words, the size of the steps taken in weight space should be chosen to minimise the value of $FE_{plat,max}$. This might require smaller steps to be taken in weight space, which will also slow down training. In some instances, larger values of $FE_{plat,max}$ may be tolerated if training time is reduced significantly as a result. It should be noted that the combination of network parameters that constitute "large" and "small" steps in weight space is highly problem dependent.

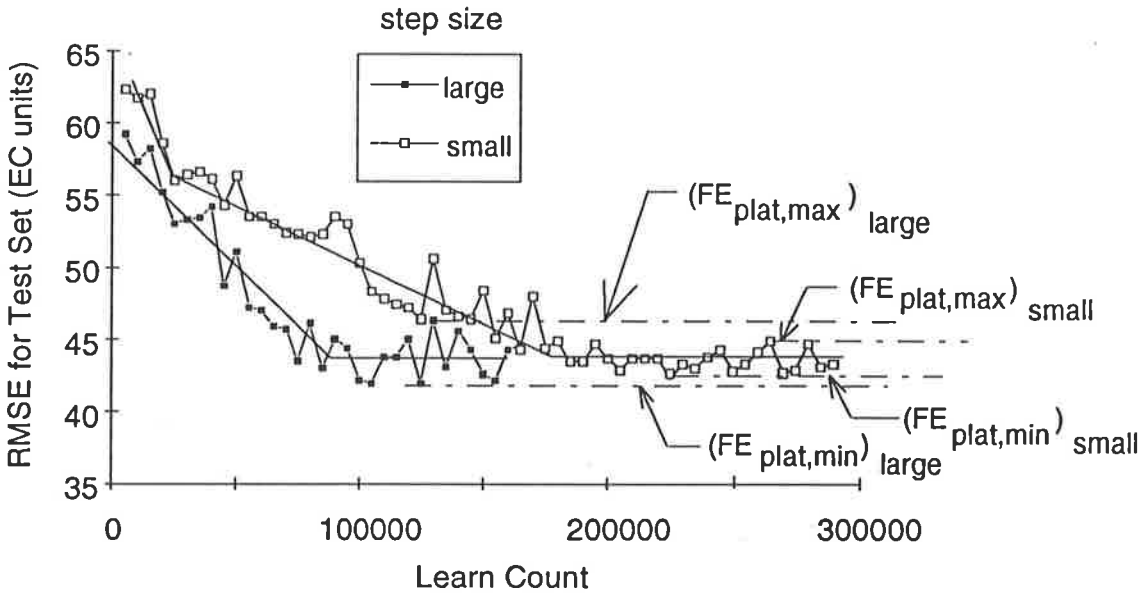


Figure 3.125: Effect of the Size of the Steps Taken in Weight Space on Generalisation Ability and Network Behaviour

In order to get an indication of the time taken to reach a local minimum in the error surface, and the magnitude of the oscillations in the RMS forecasting error once a local minimum in the error surface has been reached, for a given training / testing set and network parameters, it is suggested that at least some part of the training set should be used for independent validation in an exploratory phase. Once the exploratory phase is

complete, the network can be re-trained by presenting a fixed number of training samples to the network using the entire training set, including the data used for validation in the exploratory phase. Without the knowledge obtained in the exploratory phase, the results obtained might be far from optimal for reasons discussed above.

A knowledge of the time taken to reach a local minimum in the error surface, and of the magnitude of the oscillations in the RMS forecasting error, is also vital when comparing the performance of networks utilising different internal parameters and network geometries. Networks with different learning rates, momentums, epoch sizes etc. might be at different stages of learning, even though the same number of training samples have been presented to them (see Figure 3.125). In order to make a fair comparison between networks using different parameters and geometries, one has to ensure that both networks are in the vicinity of a local minimum at the time of testing. Even if this is the case, allowances have to be made for the oscillations in RMS forecasting error. One can only conclude that the performance of a particular network (e.g. network 1) is better than that of another network (e.g. network 2) if $(FE_{plat,max})_1 < (FE_{plat,min})_2$. Consequently, the results obtained in Sections 3.5.3.3 and 3.5.3.4 do not indicate conclusively which epoch sizes, momentums, learning rates, initial weight distributions and network geometries are optimal.

3.5.3.7 Conclusions

For the training set used, it was found that:

1. There is no advantage in using larger epoch sizes. The number of normalised weight updates required to obtain the best result was independent of epoch size. As the time taken to perform one weight update increases with increasing epoch size, learning is much faster with smaller epoch sizes. The predictive ability of the networks was found to be unaffected by epoch size.

2. Learning speed is a function of the size of the steps taken in weight space, which is increased by increasing the learning rate, the momentum, the gain of the transfer function, the error function and the epoch size (depending on the learning rule used). As a result, learning speed is affected by the combination of the above factors, and the same learning speed can be achieved by using different combinations of the above parameters.

   When small steps are taken in weight space during training, the RMSE between actual and predicted outputs decreases slowly and steadily until a local minimum in

the error surface is reached. Once the local minimum has been reached, the network basically remains at that local minimum, with only small variations in the RMS prediction error.

When bigger steps are taken in weight space during training, training is much faster. Using bigger steps in weight space generally also results in better predictions, as the network has the ability to escape, and hence find better, local minima in the error surface. However, once the network approaches a local minimum, large oscillations in the RMS prediction error occur, as the large steps taken in weight space result in the network jumping from one side of the local minimum to the other. Consequently, the result obtained is very much dependent on whether the network is tested at a time of a peak or a trough in the RMS prediction error. It is therefore necessary to test the network frequently in order to increase the chances of obtaining a good result.

If the steps taken in weight space are too large, divergent behaviour can occur.

3. Figure 3.126 shows that for the tests carried out, the behaviour of the networks was fairly controlled at learning rates of 0.005 and 0.02. The results obtained when a learning rate of 0.1 was used were clearly better than those obtained when the smaller learning rates were used. There is a large variation in the results obtained when a learning rate of 0.2 was used, indicating that the oscillations in RMS prediction error were large. Although the best result was obtained when a learning rate of 0.2 was used (in conjunction with a momentum of 0.6), it is probably preferable to use a learning rate of 0.1, as better results are obtained more consistently.

4. Figure 3.127 shows that, for the tests carried out, network performance generally increased as the momentum increased from 0.05 to 0.6. For momentums of 0.8 and 0.9, the sizes of the steps taken in weight space appear to be too large, resulting in divergent behaviour at the higher learning rates and decreased performance at the lower learning rates. Consequently, 0.6 appears to be the optimum momentum for the tests carried out.

5. Transfer and error functions have a marked impact on learning speed and predictive ability. Of the transfer and error functions tested, the hyperbolic tangent transfer function and the quadratic error function were found to perform best.

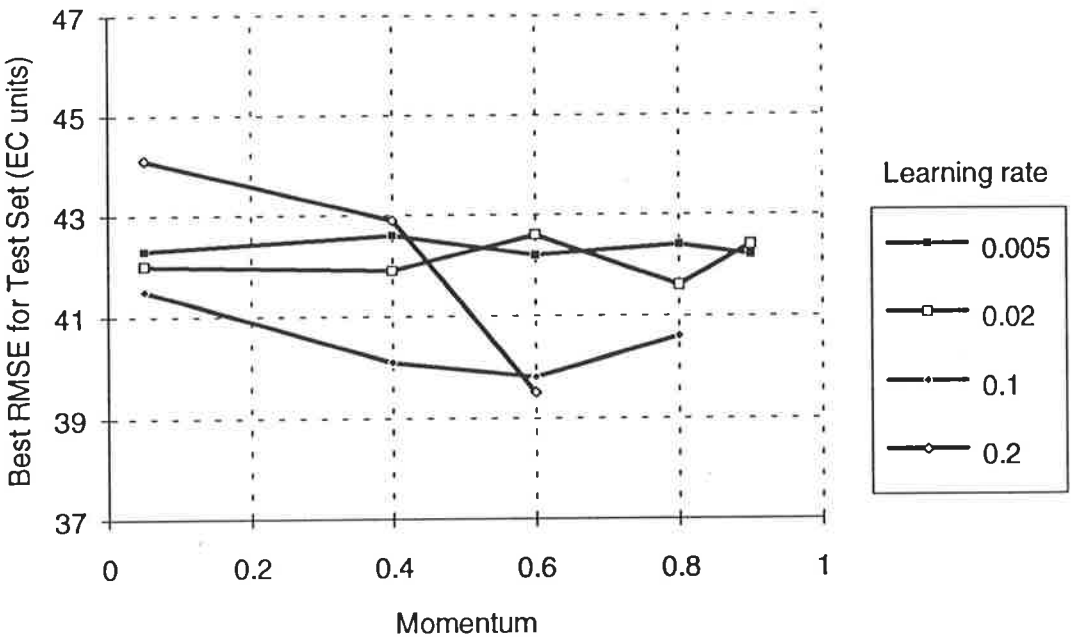


Figure 3.126: Best Results Obtained When Using Various Momentums at a Given Learning Rate

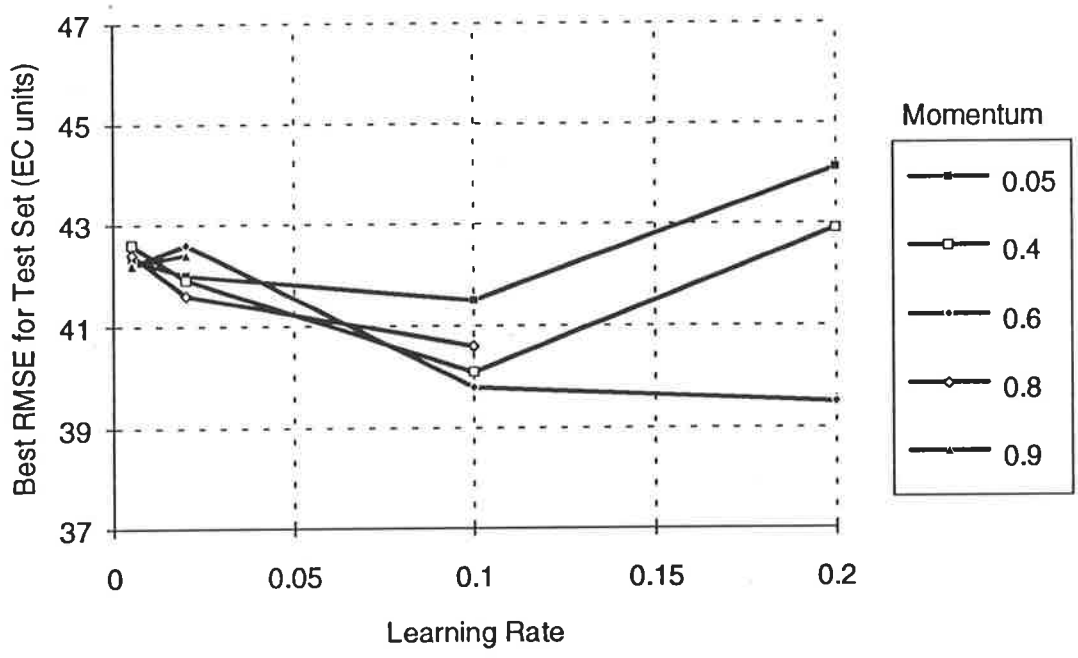


Figure 3.127: Best Results Obtained When Using Various Learning Rates at a Given Momentum

6. Commencing training from different points in weight space (i.e. different initial weight distributions) did not have any impact on learning speed and generalisation ability.

7. The ratio of first to second hidden layer nodes did not significantly affect the network's learning speed and generalisation ability. However, the best forecast was obtained when the ratio of first to second hidden layer nodes used was 3:1, which is in agreement with the results obtained by Kudrycki (1988).

8. There was a slight difference in the results obtained when different numbers of nodes were used in the first and second hidden layers, while maintaining a ratio of 3:1 between them. The predictive ability of the networks increased as the number of nodes in the first hidden layer was increased from 15 to 30 and from 30 to 60 (Figure 3.128). A slight decrease in the generalisation ability was observed as the number of nodes in the first hidden layer was increased from 60 to 90 (Figure 3.128), suggesting that the optimum number of nodes in the first and second hidden layers are 60 and 20 respectively.

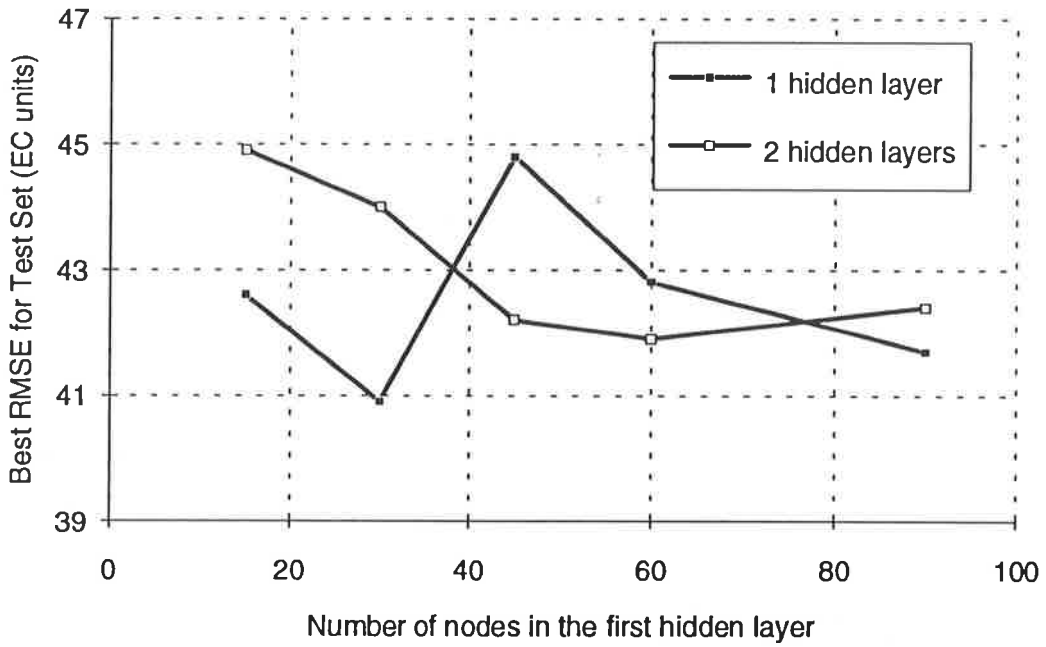


Figure 3.128: Best Results Obtained for Networks with One and Two Hidden Layers

9. When only one hidden layer was used, there was no trend relating the predictive ability of the network to the number of nodes in the hidden layer. The best forecast was obtained when 30 nodes were used in the first hidden layer (Figure 3.128).

10. As can be seen from Figure 3.128, there was greater variability in the results obtained when one hidden layer was used. However, the best 14 day forecast obtained when one hidden layer was used was better than that obtained when two

hidden layers were used. Using only one hidden layer also has the additional benefit of reduced training time.

11. Once a network was in the vicinity of a local minimum, oscillations in the RMS prediction error occurred after each weight update.

12. Using a "spiked" training set, in which all patterns to be learnt have approximately equal representation, did not improve network performance.

13. The networks used in these tests were found to have many local minima, making it difficult to obtain results close to the global minimum. This is typical for large networks. However, the following procedure was found to produce good results:
    (i) Train the network using a small epoch size, a relatively large learning rate and a relatively large momentum. This increases training speed and enables the network to escape local minima in the error surface. It should be noted that the absolute values of learning rate and momentum chosen are dependent on the nature of the error surface. The learning rate and momentum chosen should also be small enough to avoid divergent behaviour.
    (ii) Stop training at regular intervals, test the network using the testing set and calculate the RMSE between the actual and predicted values.
    (iii) Continue training until a plateau in the RMSE is reached.
    (iv) Select the network that produced the forecast with the lowest RMSE and continue training after reducing the learning rate and the momentum to very small values and setting the epoch size equal to number of samples in the training set.
    (v) Stop training after each weight update, test the network using the training set and calculate the RMSE between the actual and predicted values.
    (vi) Continue training until there is no further improvement in RMSE.

14. When using cross-validation as the stopping criterion for real time forecasting applications, it is necessary to have separate training, validation and forecasting sets. The forecasts obtained using the validation set can then be optimised using the procedure given in 13 above. The network that gives the best forecasts for the validation set can then be used for forecasting. For this method to work effectively, the validation set has to be chosen carefully, so that it is representative of the relationship to be approximated.

    When limited data are available, it is better to present a fixed number of training samples to the network during training, in order to utilise all the available data in the

training phase. However, it is vital to carry out some exploratory analysis, in which a small portion of the training set is used for validation purposes at various stages of training, in order to get an indication of how many training samples need to be presented to the network until a local minimum in the error surface is reached and the magnitude of the oscillations in the RMS forecasting error, once a local minimum in the error surface has been reached, for a particular combination of network parameters. The knowledge obtained as part of the exploratory phase (i.e. what network parameters are appropriate and how many training samples should be presented to the network) can then be used to re-train the network, using all available data for training. The knowledge of when a local minimum in the error surface has been reached may also be used to reduce the size of the steps taken in weight space at that point, in order to reduce the magnitude in the oscillations of the RMS forecasting error.

It should be noted that the results obtained apply specifically to the training set used and are by no means conclusive, as a limited number of tests were carried out.

3.5.4 Real Time Forecasts

Finally, a real time forecasting situation was simulated. When obtaining real time forecasts, the desired outputs of the forecasting set are not available during the training phase. In addition, only data prior to the period of time used for forecasting can be used for training. The last complete year of data (i.e. 1991) was used for forecasting, rather than the 12 months from July 1991 to June 1992, as all major variations in salinity occur in the middle of the year. Due to the limited amount of training data available, cross-validation was not used as a stopping criterion. Instead, a fixed number of training samples were presented to the network. The results obtained in Sections 3.5.3.3 and 3.5.3.4 indicate that:

- Network geometry does not have a significant effect on the learn count at which a local minimum in the error surface is reached.
- Training speed (i.e. at what learn count a local minimum in the error surface is reached) and the magnitude of the oscillations in the RMS forecasting error, once a local minimum in the error surface has been reached, are a function of the size of the steps taken in weight space, which is affected by the combination of learning rate, epoch size, momentum, error function and transfer function used. In the tests carried out in Section 3.5.3.3, a number of networks had reached, and remained in the vicinity of, a local minimum in the error surface at a learn count of 100,000. The internal parameters used in these tests include an epoch size of 16, the hyperbolic

tangent transfer function, the quadratic error function and a number of combinations of learning rates and momentums.

Consequently, it was decided to use an epoch size of 16, the hyperbolic tangent transfer function, the quadratic error function and to present 100,000 training samples to the networks in the real time forecasting simulation. The number of network outputs used was three (at lags 0, -4 and -13), in order to obtain the desired 1, 5 and 14 day forecasts directly. As the results obtained in Section 3.5.2.5 indicate, the number of outputs used do not significantly affect the forecasts obtained. Initially, the default values of the learning rate and momentum (Table 3.39) were used in conjunction with 45 nodes in the first hidden layer and 15 nodes in the second hidden layer (model SMB_M_A_76_12_91), to get a direct comparison with the 14 day forecast obtained using model SMB_M_A_18_10_91. The results obtained using model SMB_M_A_76_12_91 are shown in Table 3.78. The RMSE, AAPE and AAE of the 14 day forecast obtained using model SMB_M_A_76_12_91 (real time forecast) were 49.0 EC units, 6.5% and 37.0 EC units respectively, compared with an RMSE of 45.9 EC units, an AAPE of 6.5% and an AAE of 34.0 EC units obtained using model SMB_M_A_18_10_91. This indicates that model SMB_M_A_76_12_91 was able to successfully forecast salinity at Murray Bridge 14 days in advance in real time.

Table 3.78: Real Time Forecasts Obtained Using Model SMB_M_A_76_12_91 (Default Values of Learning Rate and Momentum)

Forecasting period (days)	RMSE (EC units)	AAPE (%)	AAE (EC units)
1	25.8	3.2	16.7
5	35.2	4.3	24.7
14	49.0	6.5	37.0

The results obtained in Sections 3.5.3.3 and 3.5.3.4 indicate that it is preferable to use learning rates that are smaller than the default values, in order to reduce the magnitude of the oscillations in the RMS forecasting error, once a local minimum in the error surface has been reached. Using a learning rate of 0.02 provides a good compromise between learning speed and generalisation ability, as the magnitude of the oscillation in the RMS forecasting error is relatively small, while a local minimum in the error surface is reached prior to a learn count of 100,000. Model SMB_M_A_76_12_91 was re-trained using a learning rate of 0.02 and a momentum of 0.6 (model SMB_M_A_77_12_91). The results obtained after the presentation of 100,000 training samples are shown in Table 3.79. By comparing Tables 3.78 and 3.79, it can be seen

that, as expected, the forecasts obtained using model SMB_M_A_77_12_91 are consistently better than those obtained using model SMB_M_A_76_12_91, regardless of the forecasting period or performance measure considered.

Table 3.79: Real Time Forecasts Obtained Using Model SMB_M_A_77_12_91 ($\eta = 0.02$, $\mu = 0.6$)

Forecasting period (days)	RMSE (EC units)	AAPE (%)	AAE (EC units)
1	20.5	2.7	14.2
5	30.7	3.5	20.2
14	47.5	6.4	34.5

The results obtained in Section 3.5.3.4 indicate that the number of hidden layers (i.e. either one or two) does not have a significant effect on generalisation ability for the training / testing set used. However, by omitting the second hidden layer, training speed can be increased. Model SMB_M_A_77_12_91 was re-trained using only one hidden layer with 45 nodes (model SMB_M_A_78_12_91). The results obtained are shown in Table 3.80. By comparing Tables 3.79 and 3.80, it can be seen that the forecasts obtained using a network with one hidden layer are marginally better than those obtained when a second hidden layer was used. In addition, the time taken to process 100,000 training samples was reduced from 60 to 45 minutes.

Table 3.80: Real Time Forecasts Obtained Using Model SMB_M_A_78_12_91 (One Hidden Layer of Nodes)

Forecasting period (days)	RMSE (EC units)	AAPE (%)	AAE (EC units)
1	18.5	2.5	13.9
5	29.6	3.3	21.0
14	46.9	6.1	33.2

The reliance of the model on the accuracy of the flow and river level forecasts was also investigated (model SMB_M_A_79_13_91) by omitting the flow inputs at lags -15, -13, ..., -1 and the river level inputs at lags -3 and -1 from training / testing set 12, thus reducing the total number of inputs from 51 to 39 (training / testing set 13). A learning rate of 0.02 and a momentum of 0.6 were used. The network geometry used was 39-30-0-3. The results obtained after the presentation of 100,000 training samples are shown in Table 3.81. By comparing Tables 3.80 and 3.81, it can be seen that the

forecasts obtained when the flow and river level forecasts were omitted are only slightly worse than those obtained when they were included in the training set, indicating that the accuracy of the flow and level forecasts does not have a significant impact on model performance.

Table 3.81: Real Time Forecasts Obtained Using Model SMB_M_A_79_13_91 (No Inputs of Future Flows and Levels)

Forecasting period (days)	RMSE (EC units)	AAPE (%)	AAE (EC units)
1	19.7	2.8	13.9
5	29.8	4.0	21.1
14	48.1	6.4	36.7

Plots of the 1, 5 and 14 day forecasts obtained using model SMB_M_A_79_13_91 are shown in Figures 3.129 to 3.131. It can be seen that the forecasts are good for all forecasting periods. As the forecasting length increases, a slight shift between the actual and predicted salinities starts to occur, and the model's ability to forecast the peak in the vicinity of day 200 is reduced. However, the 14 day forecasts obtained are very good, as all major variations in salinity are predicted without appreciable lag.

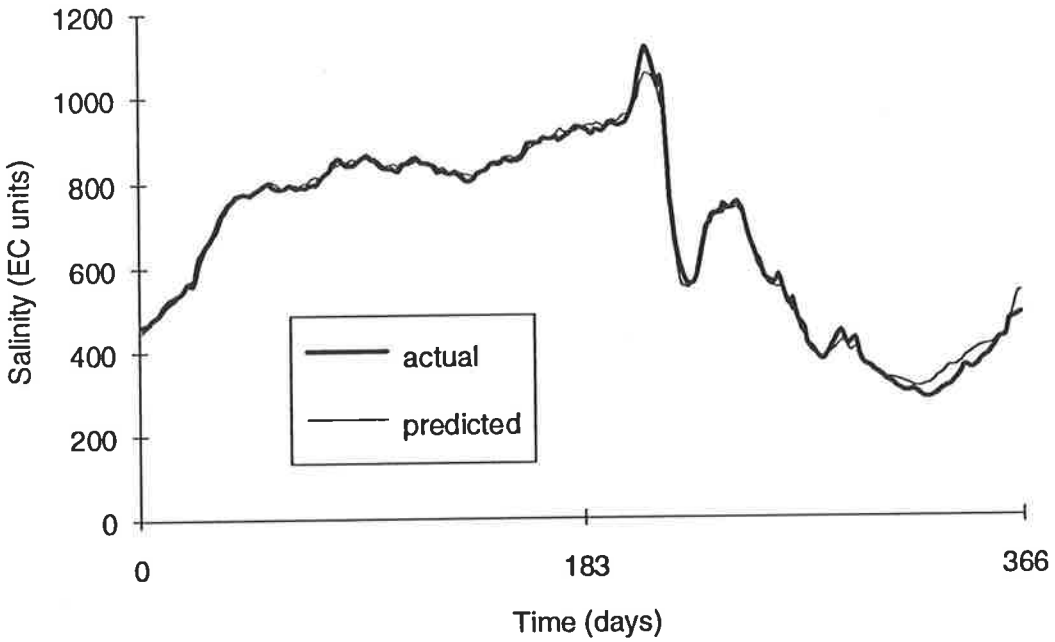


Figure 3.129: Real Time Forecast of Salinity in the River Murray at Murray Bridge Obtained Using Model SMB_M_A_79_13_91 - 1991 - 1 Day in Advance

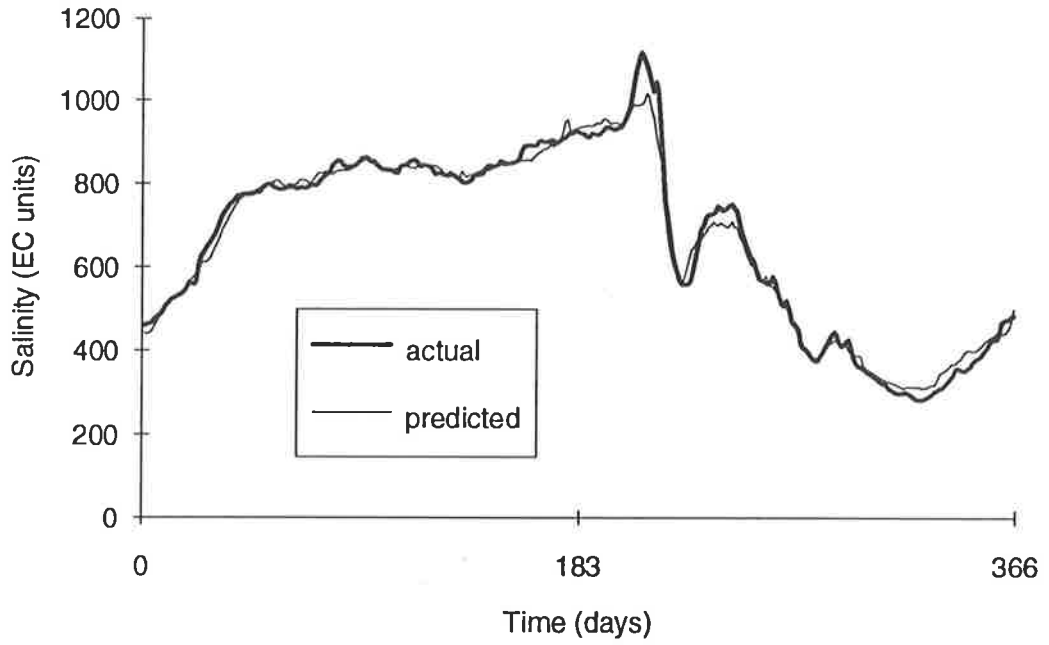


Figure 3.130: Real Time Forecast of Salinity in the River Murray at Murray Bridge Obtained Using Model SMB_M_A_79_13_91 - 1991 - 5 Days in Advance

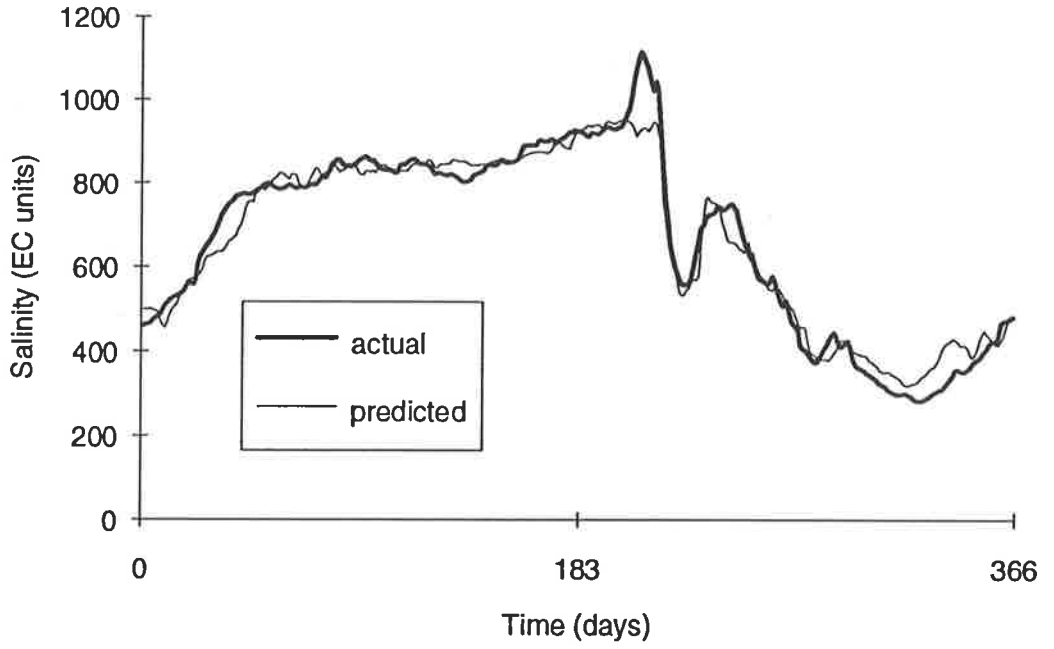


Figure 3.131: Real Time Forecast of Salinity in the River Murray at Murray Bridge Obtained Using Model SMB_M_A_79_13_91 - 1991 - 14 Days in Advance

Once a network has been trained, a general relationship between the inputs and outputs contained in the training set has been obtained. However, depending on the length and quality of the available training data, all possible input/output combinations are generally

not contained in the training set. In addition, there might be some future changes in the system that is being modelled. Consequently, for real time forecasting applications, it is advisable to re-train the model as new data become available. For the case study considered in this chapter, it would be sufficient to re-train the model once every year, as one year of data is required to adequately capture the seasonal cycle of salinity variation.

3.5.5 Summary / Conclusions

The generalisation ability of the MANN models was very good. The RMSEs and the AAPEs of the independent 14 day forecasts obtained for 1988 to 1991 ranged from 44.5 EC units to 49.3 EC units and from 5.3% to 7.0% respectively. A RMSE of 48.1 EC units and an AAPE of 6.4% were obtained in a real time forecasting simulation for 1991. The plots of the best 14 day forecasts also indicate that the forecasts are very good, as all major variations in salinity are predicted without appreciable lag. One and five day forecasts were also obtained as part of the real time forecasting simulation, which provide a basis of comparison with the forecasts obtained using the UANN, univariate and multivariate time series models.

The data used for training and those used for testing appear to have a significant effect on the results obtained. Consequently, it is vital to choose the testing set carefully. The data used for testing should be representative of the general relationships the model is trying to approximate. Training several models, and using different data for training and for testing in each, is a good way of determining the generalisation ability of a model, as it minimises the significance of choosing particular training and testing sets, while maximising the use of the available data.

Although using ANNs does not yield a direct relationship between a set of inputs and outputs, useful information about the relative strength (and sign) of the relationship between the network inputs and outputs can be obtained with the aid of a sensitivity analysis. The information obtained from such a sensitivity analysis is useful for determining the critical network inputs.

Generally, adding less significant model inputs does not appear to have a marked effect on the generalisation ability of MANN models. However, as the results in Sections 3.5.2.3 and 3.5.2.4 indicate, if the number of model inputs, and hence the size of the network, is too large, the results obtained might be adversely affected, possibly due to the difficulty in finding a good local minimum in the weight space. In addition, larger

networks also require larger training sets, in order to optimise the connection weights efficiently, and take longer to train as a result of decreased processing speed.

The results obtained in Section 3.5.2.5 indicate that network performance is not significantly affected by the number of network outputs. However, one would expect that if the number of outputs was very large, problems of finding a good local minimum in the error surface, similar to those described for large numbers of network inputs described above, would be encountered due the increased number of connection weights.

In the majority of tests carried out, the generalisation ability of the networks increased steadily, until a local minimum in the error surface was reached. Subsequently, oscillations in the RMS forecasting error occurred, the magnitude of which was a function of the size of the steps taken in weight space. Guidelines for optimising generalisation ability are summarised in Section 3.5.3.7.

The transfer and error functions had a significant effect on the generalisation ability of the networks. The hyperbolic tangent transfer function and the quadratic error function were found to perform best. The choice of the learning rate, momentum, epoch size and network geometry did not have a significant effect on generalisation ability. Training speed was greatly affected by the size of the steps taken in weight space, which is a function of the epoch size (depending on the learning rule used), the learning rate, the momentum, the error function and the gain of the transfer function. When larger steps are taken in weight space, learning speed, and the ability of networks to escape local minima in the error surface, is increased. However, the magnitude of the oscillations, once a local minium in the error surface has been reached, is also greater, which can cause problems in real time forecasting applications when limited data are available, as discussed in Section 3.5.3.6. In addition, if the step size is too large, divergent behaviour may occur or the network may cease learning.

3.6 Development of Univariate Time Series Model

3.6.1 Introduction

In this section, the development of the univariate time series (ARIMA) model for the prediction of salinity in the River Murray at Murray Bridge is described. The objective of this section is to develop an ARIMA model, which provides a basis of comparison for the multivariate time series (VARIMA) and the univariate ANN models. In order to

obtain the best possible ARIMA model, the effects of the degree of differencing and the number of AR and MA parameters were investigated.

The Box-Jenkins methodology (Section 2.2.12) was used to develop the ARIMA models. Forecasting periods of 1, 5 and 14 days were used, as was the case with the ANN models. For all models, data from 1987 to 1990 were used for training and data from 1991 were used for testing in order to obtain a real time forecast, as was the case when the univariate ANN model was developed.

3.6.2 Inspection of the Plot of the Time Series

A plot and discussion of the salinity data at Murray Bridge are given in Section 3.3.3.1. At this stage of modelling, it is assumed that the distribution of the data approaches normality. At the diagnostic checking stage, the model residuals will be examined to check whether the model is adequate. If this is found to be the case, there is no need to transform the original data using a Box-Cox transform (Verbyla, 1995).

3.6.3 Checking of the Time Series for Stationarity

The time series is non-stationary, as discussed in Section 3.4.2.

3.6.4 Transformation of Data by Differencing

The time series was differenced in Section 3.4.3. When developing the univariate ANN model, data set 11 (i.e. the data set that was differenced using d = 1 and D = 1) was used for reasons discussed in Section 3.4.3. In addition to data set 11, data set 12 (i.e. the data set that was differenced using d = 1 and D = 2) was used in the development of the ARIMA model to investigate the effect of various degrees of differencing on model performance. Data set 12 was chosen as it has the smallest number of significant values in the ACF and PACF (Section 3.4.3), and should therefore require fewer AR and MA parameters. A plot of data set 12 is shown in Figure 3.132. Plots of the ACF and the PACF, including the bounds of significance, are shown in Figures 3.133 and 3.134. It can be seen that the variance of the data does not appear to be constant, and that the values of the ACF and PACF are substantially greater than the bounds of significance in the vicinity of one seasonal lag. However, as the ACF and PACF cut off at lags less than or equal to 732, data set 12 may be deemed stationary (Bowermann and O'Connell, 1979).

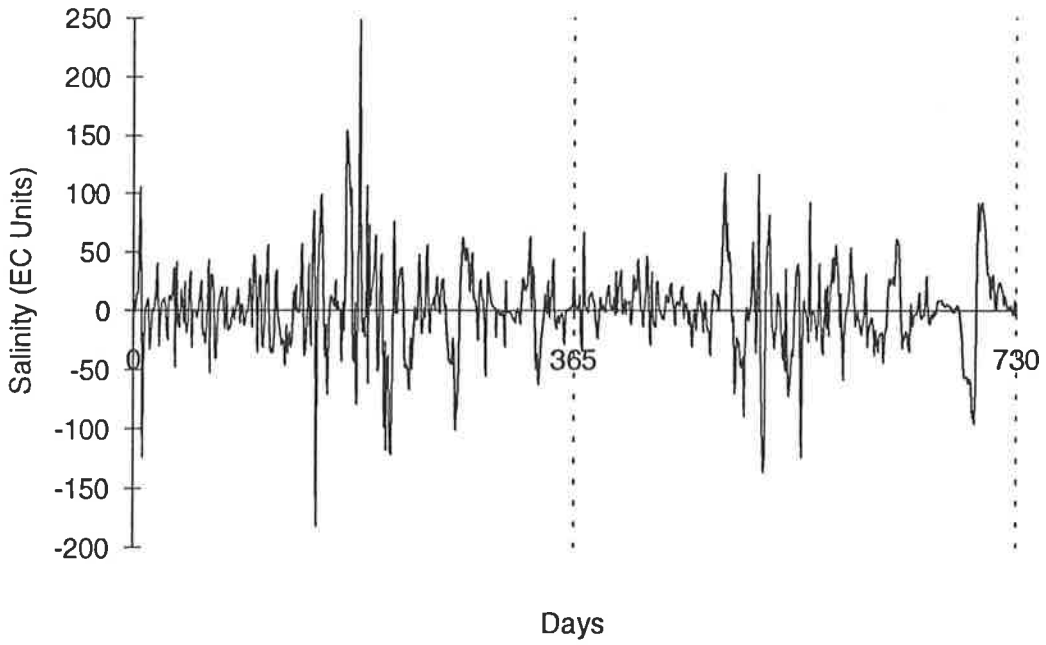


Figure 3.132: Data Set 12

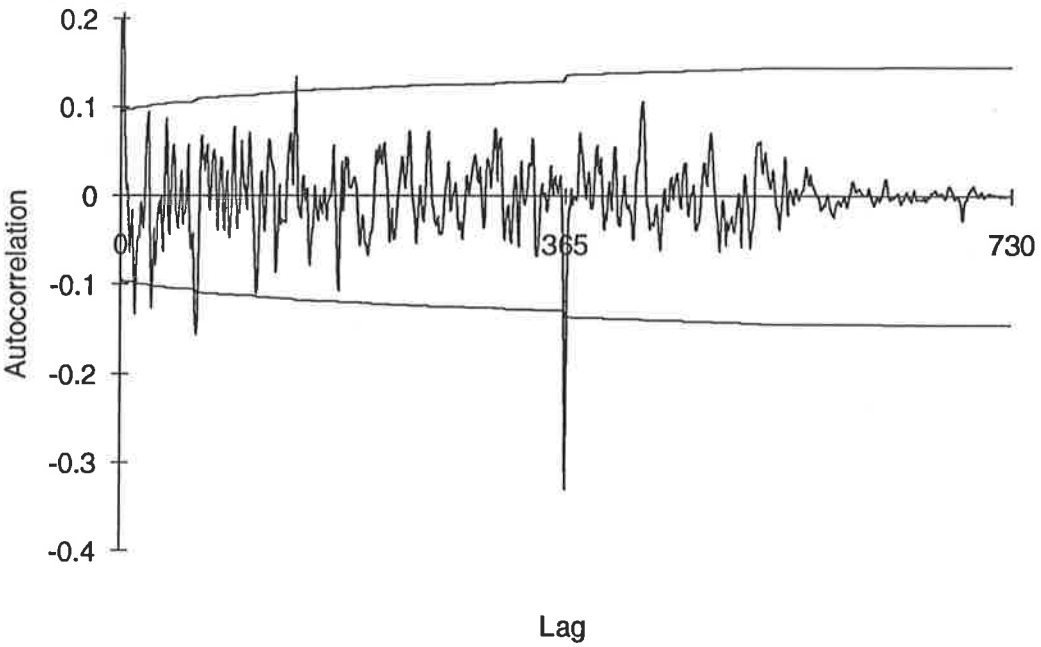


Figure 3.133: ACF for Data Set 12

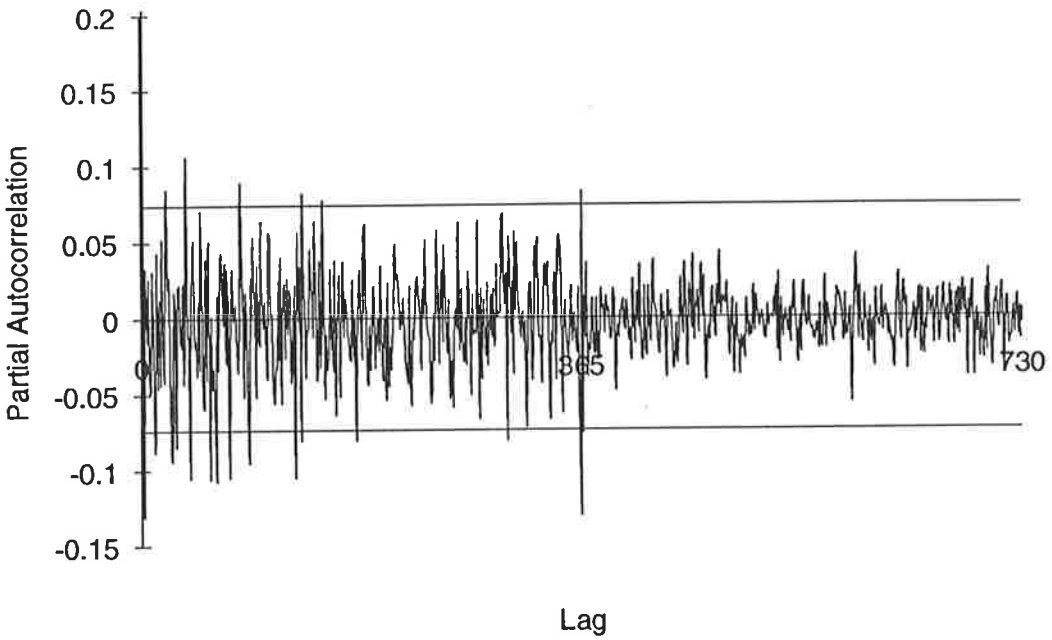


Figure 3.134: PACF for Data Set 12

3.6.5 Identification of Model Type and Order

For data sets 11 and 12, both the ACF and the PACF have a distinct cut off. As a result, either an AR or a MA model could be used. However, by using a mixed ARMA model, fewer model parameters should be required to obtain a suitable model, and hence this type of model was chosen. As discussed in Section 2.2.8.1, the order of the AR parameters required is related to the lags at which significant values occur in the PACF, and the order of the MA parameters required is related to the lags at which significant values occur in the ACF. However, as discussed by Chatfield (1975), even if a time series is completely random, 1 out of 20 autocorrelation values will generally be significant. It is therefore difficult to determine which of the significant values represent strongly correlated values, especially since most of the significant values are only slightly greater than the bounds of significance. For this reason, it was decided to adopt an iterative approach. The following procedure was used to determine the order of the MA and AR parameters:

1. Obtain the preliminary parameter estimates for an AR model and a MA model, each with up to 732 parameters.
2. Determine which of the parameters are significant (i.e. which parameters are greater than or equal to $2/\sqrt{n}$).
3. Delete all parameters that are found to be insignificant.
4. Obtain preliminary parameter estimates for a mixed ARMA model using the parameters remaining after step 3.

5. Repeat steps 2 to 4 until all the remaining parameters are significant.

The above procedure was applied to data sets 12 and 11. The preliminary parameter estimates were obtained with the aid of computer program 'PRELIM'. It was found that a number of significant AR and MA parameters were only marginally greater than $2/\sqrt{n}$, and very few parameters were significantly greater than $2/\sqrt{n}$.

Final parameter estimation was carried out for a number of models, including those which incorporated all significant parameters (e.g. models SMB_U_T_05_12_91 and SMB_U_T_07_12_91), and models which only incorporated those parameters significantly greater than $2/\sqrt{n}$ (e.g. models SMB_U_T_08_12_91 and SMB_U_T_01_11_91). Details of all the models used are given in Tables 3.82 and 3.83.

It should be noted that a model containing AR parameters of order greater than 365, as well as MA parameters of order greater than 365, could not be developed for data set 12, as the sum of the maximum order of AR parameters and the maximum order of MA parameters has to be less than the total number of values in the data set (i.e. 730). However, the same restriction does not apply to data set 11 (see model SMB_U_T_01_11_91), as the number of values lost is less as a result of the lesser degree of seasonal differencing.

Table 3.82: Number and Order of MA and AR Parameters for the Models using Data Set 11

Model	AR Parameters		MA Parameters	
	No.	Order	No.	Order
SMB_U_T_01_11_91	3	1, 365, 366	4	364, 365, 366, 367
SMB_U_T_02_11_91	3	1, 365, 366	5	1, 364, 365, 366, 367
SMB_U_T_03_11_91	3	1, 365, 366	1	1
SMB_U_T_04_11_91	3	1, 365, 366	0	

Table 3.83: Number and Order of MA and AR Parameters for the Models using Data Set 12

Model	AR Parameters		MA Parameters	
	No.	Order	No.	Order
SMB_U_T_05_12_91	1	1	28	1, 2, 3, 12, 16, 23, 24, 26, 27, 36, 39, 62, 63, 64, 65, 113, 144, 180, 365, 366, 367, 377, 394, 427, 428, 429, 491, 524
SMB_U_T_06_12_91	5	1, 2, 3, 365, 366	1	1
SMB_U_T_07_12_91	22	1, 2, 3, 17, 24, 37, 38, 40, 41, 53, 54, 57, 58, 63, 73, 74, 116, 117, 132, 133, 365, 366	1	1
SMB_U_T_08_12_91	3	1, 365, 366	1	1
SMB_U_T_09_12_91	2	1, 365	1	1
SMB_U_T_10_12_91	3	1, 365, 366	2	1, 23

3.6.6 Estimation of Model Parameters

The final least-squares parameter estimates were obtained with the aid of computer program 'PAREST'. All initial parameter estimates were chosen to be 0.1 in accordance with the recommendation made by Bowermann and O'Connell (1979). Final parameter estimates were obtained for all the models listed in Tables 3.82 and 3.83. These estimates are summarised in Tables 3.84 to 3.93.

The final estimates of the model parameters in Tables 3.84 to 3.93 show that the parameters at lag 1 and the parameters in the vicinity of one seasonal lag (i.e. 364, 365, 366 and 367) are clearly dominant. These lags coincide with the lags at which the largest spikes occur in the ACF and the PACF. This suggests that only the dominant spikes (i.e. spikes that are considerably greater than the bounds of significance) in the ACF and in the PACF represent strongly correlated values, and that the significant values that are close to the boundaries of significance may be ignored.

Table 3.84: Final Estimates of Model Parameters for Model SMB_U_T_01_11_91

AR Parameters		MA Parameters	
Order	Parameter	Order	Parameter
1	+0.6262	364	+0.1363
365	-0.4056	365	-0.1381
366	+0.2832	366	+0.1487
		367	+0.1151

Table 3.85: Final Estimates of Model Parameters for Model SMB_U_T_02_11_91

AR Parameters		MA Parameters	
Order	Parameter	Order	Parameter
1	+0.5833	1	-0.0704
365	-0.4058	364	+0.1376
366	+0.2660	365	-0.1254
		366	+0.1427
		367	+0.1297

Table 3.86: Final Estimates of Model Parameters for Model SMB_U_T_03_11_91

AR Parameters		MA Parameters	
Order	Parameter	Order	Parameter
1	+0.5989	1	-0.0421
365	-0.3791		
366	+0.2145		

Table 3.87: Final Estimates of Model Parameters for Model SMB_U_T_04_11_91

AR Parameters		MA Parameters	
Order	Parameter	Order	Parameter
1	+0.6240		
365	-0.3799		
366	+0.2253		

Table 3.88: Final Estimates of Model Parameters for Model SMB_U_T_05_12_91

AR Parameters		MA Parameters	
Order	Parameter	Order	Parameter
1	+0.0432	1	-0.1574
		2	+0.0031
		3	+0.0189
		12	+0.0918
		16	+0.1138
		23	-0.1121
		24	-0.0723
		26	+0.1184
		27	+0.1146
		36	+0.1169
		39	-0.0790
		62	+0.0957
		63	+0.0681
		64	+0.1325
		65	+0.0787
		113	-0.0202
		144	-0.1264
		180	+0.0947
		365	+0.4892
		366	+0.1004
		367	+0.0426
		377	-0.1044
		394	-0.0353
		427	-0.0514
		428	-0.0465
		429	-0.0837
		491	+0.0737
		524	-0.0752

Table 3.89: Final Estimates of Model Parameters for Model SMB_U_T_06_12_91

AR Parameters		MA Parameters	
Order	Parameter	Order	Parameter
1	+0.6384	1	-0.1987
2	+0.0952		
3	-0.0774		
365	-0.5168		
366	+0.3370		

Table 3.90: Final Estimates of Model Parameters for Model SMB_U_T_07_12_91

AR Parameters		MA Parameters	
Order	Parameter	Order	Parameter
1	+0.6452	1	-0.1798
2	+0.0754		
3	-0.0811		
17	+0.0060		
24	-0.0366		
37	+0.0442		
38	-0.0722		
40	-0.0435		
41	+0.0323		
53	+0.0054		
54	+0.0065		
57	-0.0063		
58	-0.0004		
63	-0.0274		
73	+0.0416		
74	+0.0265		
116	+0.0257		
117	-0.0296		
132	-0.0730		
133	+0.0453		
365	-0.5243		
366	+0.3219		

Table 3.91: Final Estimates of Model Parameters for Model SMB_U_T_08_12_91

AR Parameters		MA Parameters	
Order	Parameter	Order	Parameter
1	+0.6977	1	-0.1284
365	-0.5136		
366	+0.3492		

Table 3.92: Final Estimates of Model Parameters for Model SMB_U_T_09_12_91

AR Parameters		MA Parameters	
Order	Parameter	Order	Parameter
1	+0.2669	1	-0.4547
365	-0.5135		

Table 3.93: Final Estimates of Model Parameters for Model SMB_U_T_10_12_91

AR Parameters		MA Parameters	
Order	Parameter	Order	Parameter
1	+0.6966	1	-0.1457
365	-0.5119	23	-0.1585
366	+0.3419		

3.6.7 Diagnostic Checking

A plot of the residuals (not shown here) showed that there were no outliers and no apparent trend. In addition, the mean of the residuals was zero. The variance of the residuals was found to be constant, and the histogram of residual frequencies (not shown here) was normally distributed, indicating that there is no need to transform the original data using a Box-Cox transform (Hipel, 1985).

The Box-Pierce method was then used to examine the residuals. The ACF and the PACF of the residuals (not shown here) were calculated using computer program 'APCORR' and checked for any significant spikes. There were no significant spikes at low lags (e.g. 1 to 10) and no significant spikes at seasonal lags (e.g. 360 to 370), suggesting that the residuals are uncorrelated. There were occasional significant values,

all of which were close to the bounds of significance. As discussed in Section 3.6.6, such values do not suggest strong correlations and may be ignored.

The Portmanteau lack-of-fit test was also used to check the whiteness of the residuals. The Box-Pierce chi-square statistic, Q^{BP} (Equation 2.78), is the most well known test statistic for the Portmanteau lack-of-fit test and was used here. Q^{BP} is evaluated as part of computer program 'APCORR'. Bowermann and O'Connell (1979) suggest that a value of k' = 3L should be used. Due to the limited number of residuals available (< 3L), a value of k' = 365 was chosen. This is conservative, as the gap between Q^{BP} and $\chi^2_{(5)}$ generally widens with an increase in k'. It should be noted that a value of k' = 363 had to be used for models SMB_U_T_06_12_91 to SMB_U_T_10_12_91, as the maximum number of residuals available was 364. The values of Q^{BP} and $\chi^2_{(5)}$ for the various models are shown in Tables 3.94 and 3.95. $\chi^2_{(5)}$ was evaluated using the following equation (Kreyszig, 1979):

$$\chi^2_{(5)} = 0.5\ (\sqrt{(2m' - 1)} + 1.64)^2 \qquad (3.6)$$

where m' = the number of degrees of freedom

Table 3.94: Values of Q^{BP} and $\chi^2_{(5)}$ for the Models using Data Set 11

Model	m'	Q^{BP}	$\chi^2_{(5)}$
SMB_U_T_01_11_91	358	296	403
SMB_U_T_02_11_91	357	295	402
SMB_U_T_03_11_91	361	329	406
SMB_U_T_04_11_91	362	327	407

Table 3.95: Values of Q^{BP} and $\chi^2_{(5)}$ for the Models using Data Set 12

Model	m'	Q^{BP}	$\chi^2_{(5)}$
SMB_U_T_05_12_91	336	205	379
SMB_U_T_06_12_91	357	163	402
SMB_U_T_07_12_91	340	151	384
SMB_U_T_08_12_91	359	168	404
SMB_U_T_09_12_91	360	265	405
SMB_U_T_10_12_91	358	149	403

As can be seen from Tables 3.94 and 3.95, Q^{BP} is less than $\chi^2_{(5)}$ in all cases, indicating that all models are adequate and can be used for forecasting.

3.6.8 Forecasting

Computer program 'FCAST' was used to produce the multi-step forecasts and to convert the differenced forecasts back to their original values. The 1, 5 and 14 day forecasts were examined to assess each model's short-term and longer-term forecasting ability.

For data set 11, the differenced forecasts (u_t) were converted to their original (non-differenced) form (z_t) using Equation 3.2. For data set 12, the following equation was used to carry out the conversions:

$$z_t = u_t + z_{t-1} + 2\,z_{t-365} - 2\,z_{t-366} - z_{t-730} + z_{t-731} \tag{3.7}$$

For each of the above conversions, the term z_{t-1} is required. However, for forecasts greater than one day ahead, this value is not available. Consequently, a predicted value has to be used. This value can be obtained by using a recursive procedure, which involves the following steps (note: the example given is for data set 11; the same procedure can be applied to data set 12 by using Equation 3.7):

- The one day forecast is obtained in accordance with the following equation (It should be noted that z_t is the last known value of z):

$$z_{t+1} = u_{t+1} + z_t + z_{t-364} - z_{t-365} \tag{3.8}$$

- The two day forecast is then obtained in accordance with the following equation:

$$z_{t+2} = u_{t+2} + z_{t+1} + z_{t-363} - z_{t-364} \tag{3.9}$$

  where z_{t+1} is the one day forecast obtained in the previous step.

- This recursive procedure is repeated until the forecast of the desired forecasting length is obtained.

3.6.9 Results / Discussion

The RMSE, AAPE and AAE between actual and predicted salinities were calculated for the 1, 5 and 14 day forecasts for each of the models. These are shown in Tables 3.96 and 3.97.

Table 3.96: Results for Data Set 11

Model	Forecast period (days)	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_T_01_11_91	1	10.5	1.1	6.7
	5	37.8	4.2	24.9
	14	109.0	11.8	71.2
SMB_U_T_02_11_91	1	10.9	1.1	6.7
	5	38.1	4.2	25.0
	14	109.8	11.9	71.4
SMB_U_T_03_11_91	1	10.7	1.1	6.9
	5	37.8	4.2	24.7
	14	108.9	12.1	71.2
SMB_U_T_04_11_91	1	10.5	1.1	6.8
	5	37.7	4.2	24.8
	14	108.5	12.1	70.9

Table 3.97: Results for Data Set 12

Model	Forecast period (days)	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_U_T_05_12_91	1	34.9	3.7	22.2
	5	153.0	17.2	104.0
	14	442.1	51.2	307.5
SMB_U_T_06_12_91	1	17.3	1.7	11.4
	5	86.8	8.7	58.4
	14	257.3	25.7	173.3
SMB_U_T_07_12_91	1	18.0	1.8	12.1
	5	94.3	10.1	65.8
	14	282.5	31.0	201.1
SMB_U_T_08_12_91	1	16.7	1.7	11.2
	5	86.7	8.6	57.7
	14	257.4	25.6	172.2
SMB_U_T_09_12_91	1	28.8	2.8	18.2
	5	102.1	10.2	67.2
	14	273.0	27.7	182.4
SMB_U_T_10_12_91	1	18.5	2.1	13.0
	5	112.5	12.4	78.3
	14	351.3	38.1	240.3

The following general comments can be made by examining the prediction errors and plots of forecasts for the various models:

1. As expected, the forecast accuracy decreases as the forecast length increases. As can be seen from Tables 3.96 and 3.97, all one day forecasts are very good. This is illustrated in Figures 3.135 to 3.137. However, any noise present is amplified as the forecast length is increased. This is demonstrated in Figures 3.137 to 3.139 and is typical for all models. The reason for this is that previous forecasts are used to obtain current forecasts (e.g. the forecast at day one is used to obtain the forecast at day 2; the forecasts at days 1 and 2 are used to obtain the forecast at day 3 etc.).

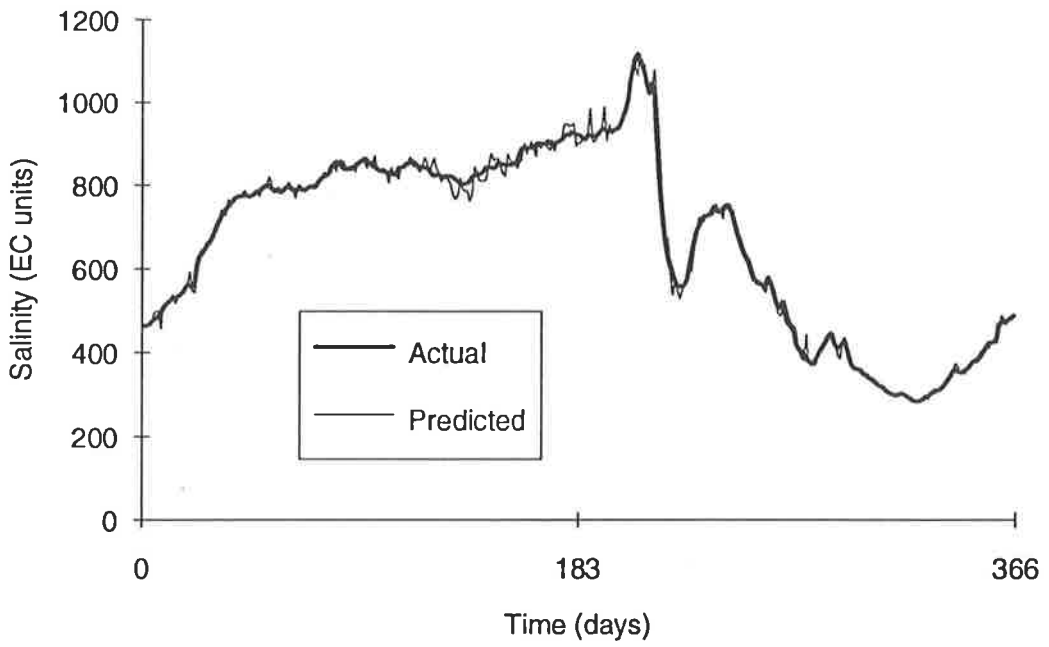


Figure 3.135: Actual and Predicted Salinities - Model SMB_U_T_06_12_91 - 1 Day in Advance

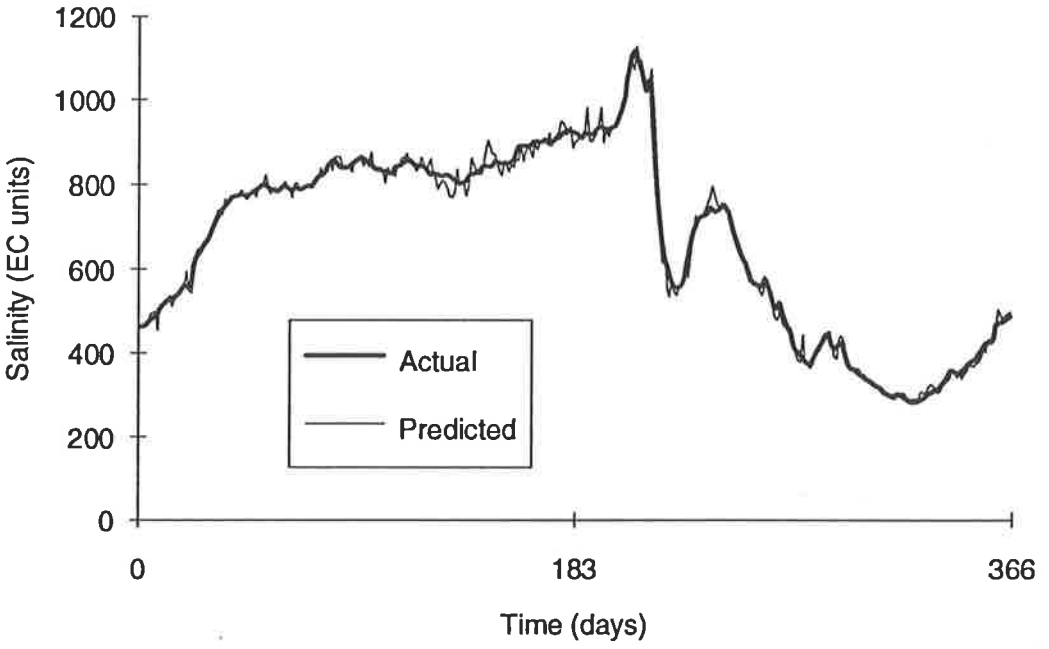


Figure 3.136: Actual and Predicted Salinities - Model SMB_U_T_10_12_91 - 1 Day in Advance

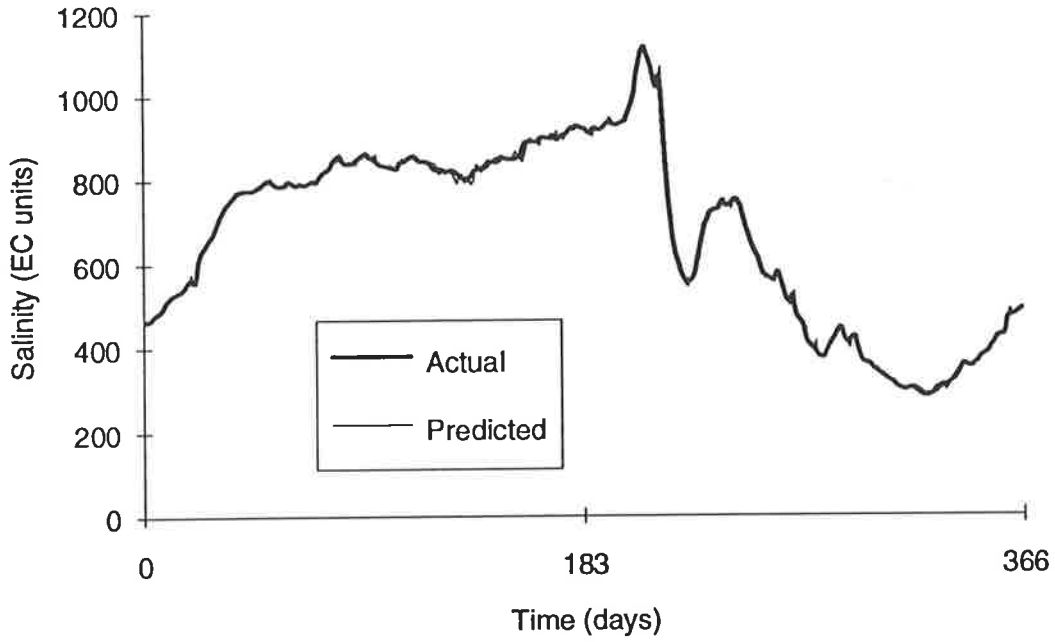


Figure 3.137: Actual and Predicted Salinities - Model SMB_U_T_03_11_91 - 1 Day in Advance

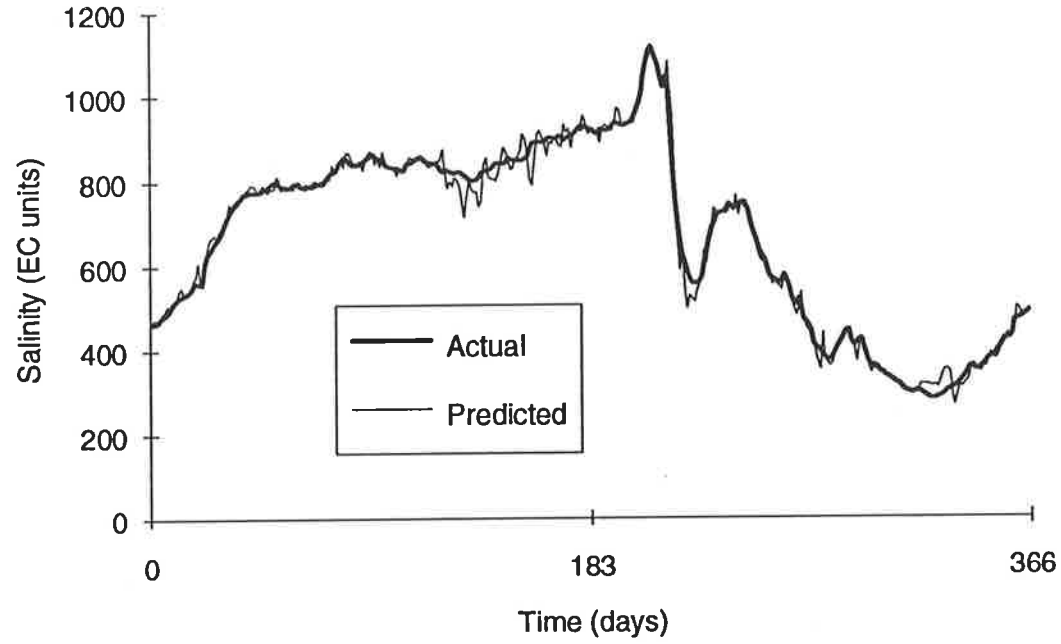


Figure 3.138: Actual and Predicted Salinities - Model SMB_U_T_03_11_91 - 3 Days in Advance

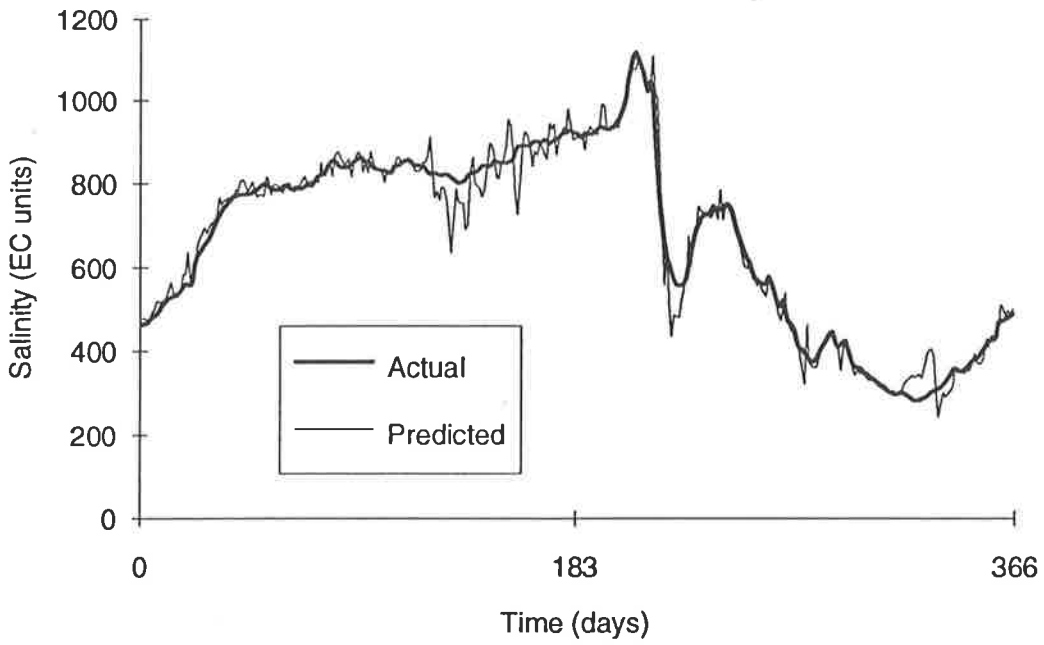


Figure 3.139: Actual and Predicted Salinities - Model SMB_U_T_03_11_91 - 5 Days in Advance

2. The presence of less-significant parameters induces noise, as illustrated in Table 3.98. Although using more parameters may produce comparable one day forecasts, undesirable noise is amplified for longer term forecasts. Using fewer parameters tends to have an averaging effect, which is desirable for longer term forecasts.

Table 3.98: Effect of Less-Significant Parameters on Model Performance

Model	No. of AR Parameters	No. of MA Parameters	RMSE (EC units)		
			1-day forecast	5-day forecast	14-day forecast
SMB_U_T_08_12_91	3	1	16.7	86.7	257.4
SMB_U_T_07_12_91	22	1	18.0	94.3	282.5

3. AR parameters at lags 1, 365 and 366 appear to be dominant. Varying the number of MA parameters does not appear to have a significant impact on model performance. Models SMB_U_T_01_11_91 to SMB_U_T_04_11_91 all have AR parameters at lags 1, 365 and 366, but have varying MA parameters. However, the performance of each model is very similar, as shown in Table 3.96. On the other hand, omitting one of the dominant AR parameters decreases prediction accuracy substantially, as illustrated by models SMB_U_T_08_12_91 and SMB_U_T_09_12_91. The AR parameter at lag

366 was omitted for model SMB_U_T_09_12_91 resulting in a marked increase in forecasting error, which can be seen by inspecting Table 3.97 and by comparing Figures 3.140 and 3.141.

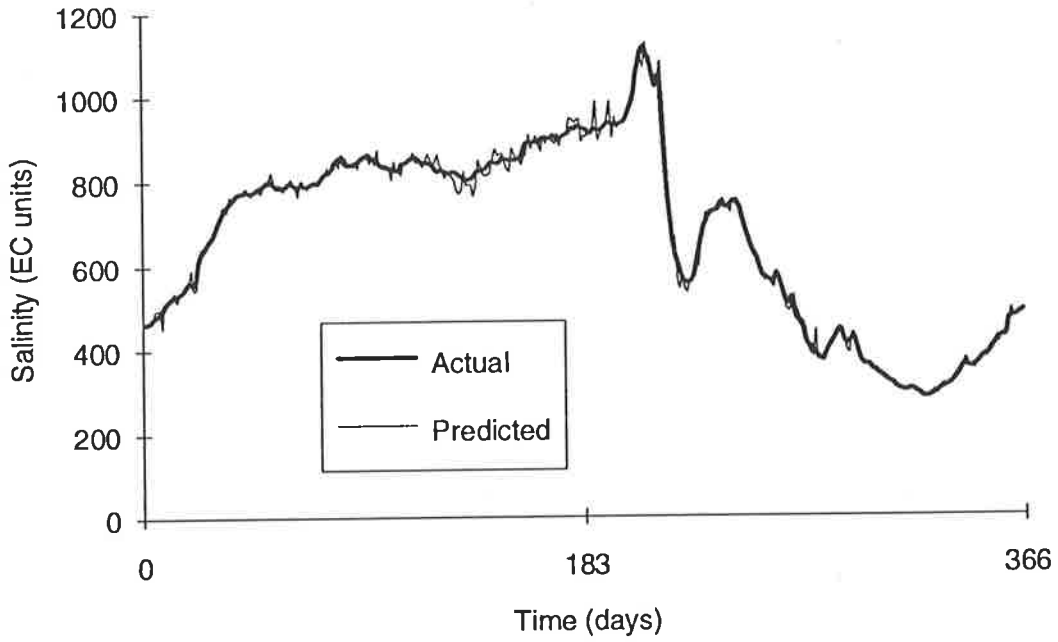


Figure 3.140: Actual and Predicted Salinities - Model SMB_U_T_08_12_91 - 1 Day in Advance (AR Parameters of Order 1, 365 and 366 & MA Parameter of Order 1)

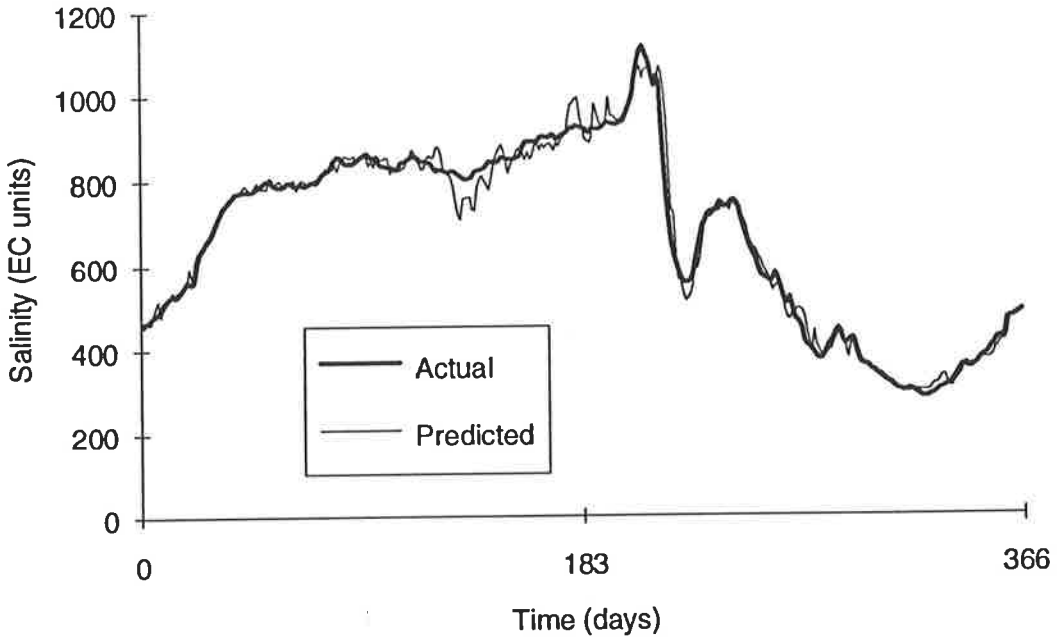


Figure 3.141: Actual and Predicted Salinities - Model SMB_U_T_09_12_91 - 1 Day in Advance (AR Parameters of Order 1 and 365 & MA Parameter of Order 1)

4. Models using data set 11 perform better than those using data set 12. This indicates that higher order differencing introduces additional noise. This is clearly shown in Tables 3.96 and 3.97 as well as Figures 3.142 and 3.143 for models SMB_U_T_04_11_91 and SMB_U_T_08_12_91, the best models for the respective data sets.

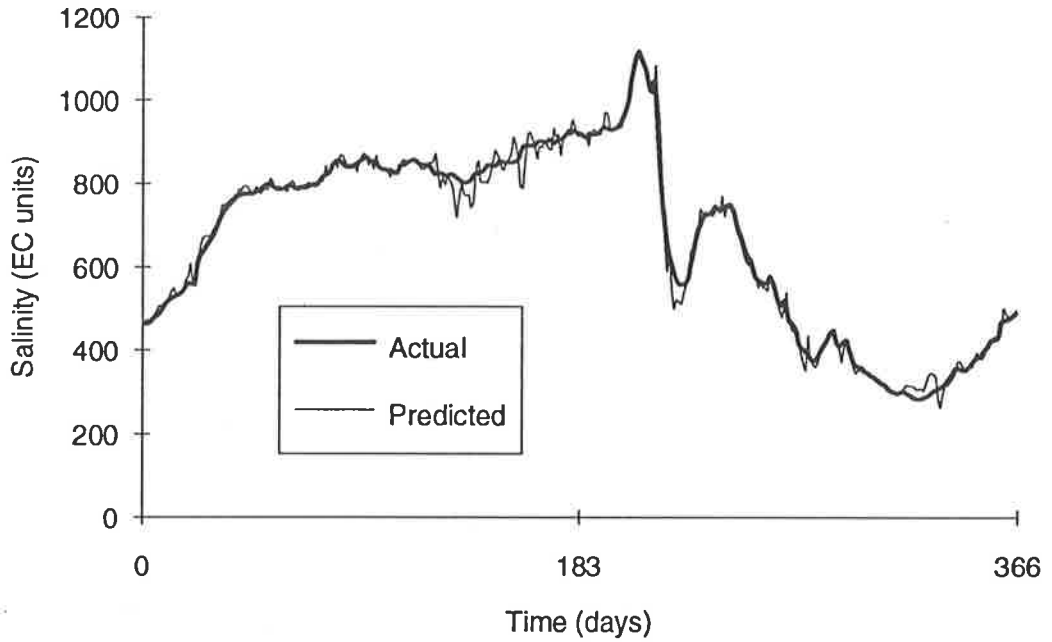


Figure 3.142: Actual and Predicted Salinities - Model SMB_U_T_04_11_91 - 3 Days in Advance (Differencing of Order D = 1)

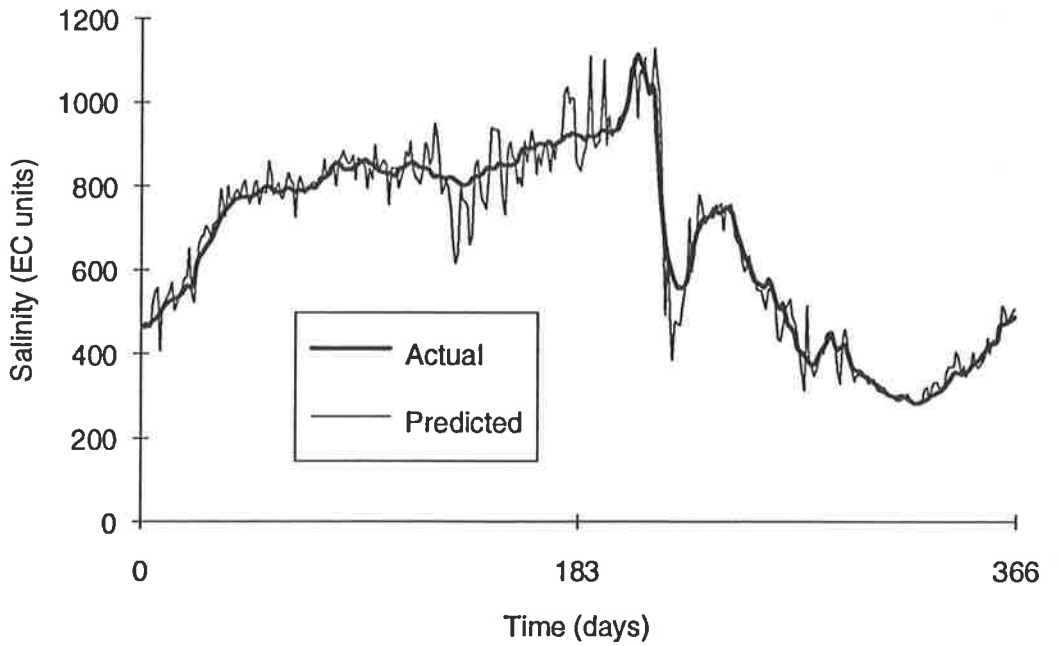


Figure 3.143: Actual and Predicted Salinities - Model SMB_U_T_08_12_91 - 3 Days in Advance (Differencing of Order D = 2)

5. The Box-Pierce chi-square statistic indicates that the models incorporating AR parameters at lags 1, 365 and 366 produce a better fit. The same models also result in the best predictions, as discussed above. On the other hand, the Box-Pierce chi-square statistic also indicates that models using more parameters, as well as models using data set 12, produce a better fit. However, as discussed above, using too many parameters can result in overfitting, which introduces noise and produces poor longer term forecasts. This indicates that although the Box-Pierce chi-square statistic is a valuable tool in determining whether a particular model is adequate, it does not necessarily indicate which model produces the best forecasts. This is particularly true when the data are noisy and when a limited data set is available.

6. A plot of the best 14-day forecast (model SMB_U_T_04_11_91) is shown in Figure 3.144. As can be seen, the model predicts the major variations in salinity reasonably well. However, there is considerable noise, resulting in very large fluctuations in predicted salinity. This reduces the usefulness of the model, as it is impossible to discern which of the predicted variations in salinity actually occur. It is interesting to note that the sharp drop in salinity is predicted with very little lag compared to the actual drop. The marked drop in predicted salinity in the vicinity of day 125 can be attributed to a sharp drop in the salinity at the same time in the previous year.

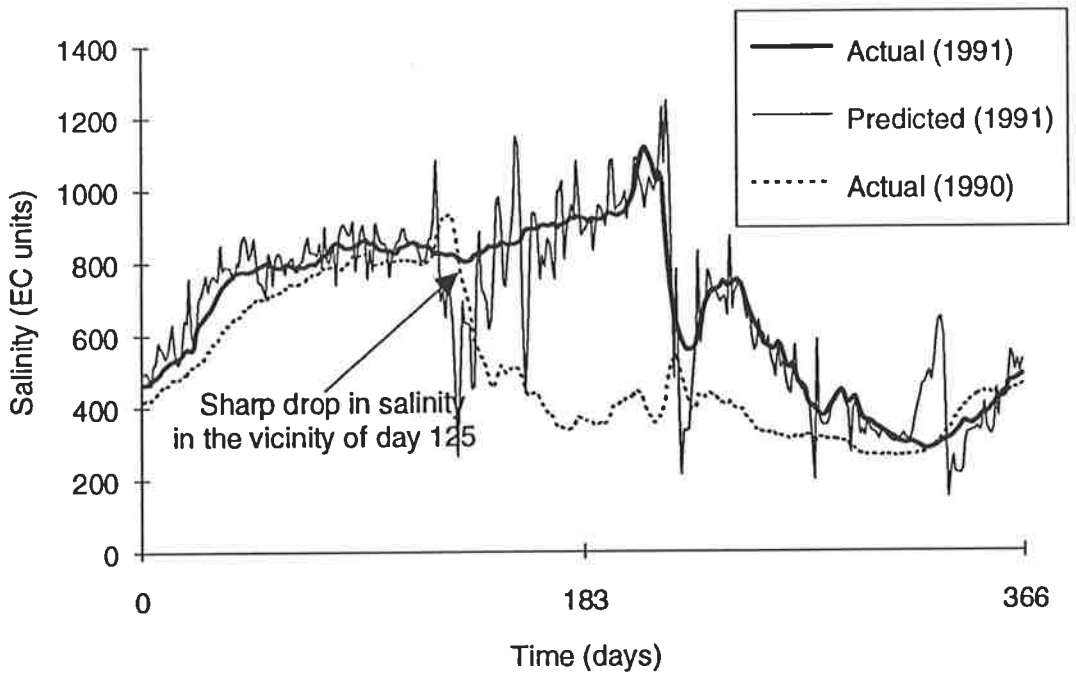


Figure 3.144: Actual and Predicted Salinities - Model SMB_U_T_04_11_91 - 14 Days in Advance

7. As discussed above, the 14 day forecasts produced are very noisy. It is possible to remove some of that noise by filtering. Filtering was carried out by applying a linear filter to the 14 day forecast produced by model SMB_U_T_04_11_91. Initially, the usefulness of filtering was assessed by using a set of symmetric weights. The weights chosen were the successive terms in the expansion $(a + b)^{2\xi}$, where a = 1/2 and b = 1/2, and are summarised in Table 3.99 (e.g. for $\xi = 1$, z_t(filtered) = $(1/4)(z_{t+1}) + (1/2)(z_t) + (1/4)(z_{t-1})$). It should be noted that the larger the value of ξ, the greater the number of values that are "lost" as part of the filtering process. The original, non-filtered, values were used in place of the "lost" values in order to compare actual values with the filtered forecasts for the whole of 1991.

Table 3.99: Filter Weights for Various Values of ξ (Symmetric Filter)

ξ	Weights
1	1/4, 1/2, 1/4
2	1/16, 4/16, 6/16, 4/16, 1/16
4	1/256, 8/256, 28/256, 56/256, 70/256, 56/256, 28/256, 8/256, 1/256

The results obtained when filtering the 14 day forecasts obtained using model SMB_U_T_04_11_91 are shown in Table 3.100. As can be seen, filtering reduces the forecasting errors and appears to be a worthwhile method for improving forecasts.

Table 3.100: Filtered 14 Day Forecasts for Model SMB_U_T_04_11_91 (Symmetrical Filter)

ξ	RMSE (EC units)	AAPE (%)	AAE (EC units)
0	108.5	12.1	70.9
1	95.0	10.7	61.8
2	89.2	10.2	58.1
4	82.3	9.5	53.8

When using filtering as a means of improving on-line forecasts, a symmetrical filter cannot be used. A one-sided filter has to be used, as only past values of the time series are available. The accuracy of using a one-sided filter was assessed, and the weights for this filter were obtained by excluding the weights of the symmetrical filter that apply to future values of the time series, and scaling the remaining weights so that their sum was equal to one. The resulting weights are shown in Table 3.101 (e.g. for $\xi = 2$, z_t(filtered) = $(6/11)(z_t) + (4/11)(z_{t-1}) + (1/11)(z_{t-2})$).

Table 3.101: Filter Weights for Various Values of ξ (One-sided Filter)

ξ	Weights
1	1/3, 2/3
2	1/11, 4/11, 6/11
4	1/163, 8/163, 28/163, 56/163, 70/163

The results obtained when applying the one-sided filter to the 14 day forecasts obtained using model SMB_U_T_04_11_91 are shown in Table 3.102. It can be seen that the one sided filter is less effective than the symmetrical filter. However, using the one-sided filter still results in a marked improvement in the results compared to using no filter at all. A comparison of the plots of the filtered and unfiltered forecasts (Figures 3.144 and 3.145) shows that the filtered 14 day forecast is considerably smoother. However, the forecast is still noisy and hence it is difficult to discern which of the predicted variations in salinity actually occur.

Table 3.102: Filtered 14 Day Forecasts for Model SMB_U_T_04_11_91 (One-sided Filter)

ξ	RMSE (EC units)	AAPE (%)	AAE (EC units)
0	108.5	12.1	70.9
1	100.9	11.2	65.3
2	97.4	10.7	61.9
4	93.0	10.2	58.5

It has been shown that filtering can be used to improve the 14 day forecasts obtained, but not to the extent that the forecast becomes useful. It should be noted, however, that only one type of filter was used here to demonstrate the effect that filtering has on the results. The results can be further improved by applying more, or possibly different types of, filters.

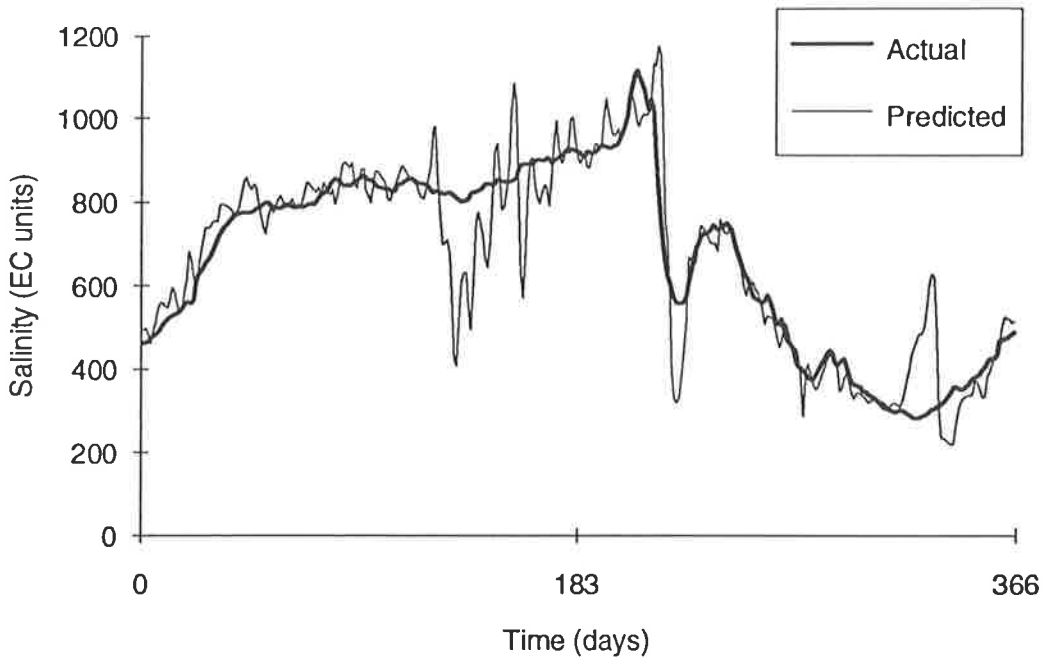


Figure 3.145: Actual and Predicted Salinities (Filtered) - Model SMB_U_T_04_11_91 - 14 Days in Advance ($\xi = 4$)

3.6.10 Summary / Conclusions

The ARIMA models developed were found to be sensitive to the number of model parameters as well as the degree of differencing used. The presence of less-significant model parameters and a high degree of differencing increased model noise, resulting in decreased performance. The best forecasts were obtained by using data set 11 (i.e d = 1 and D = 1) in conjunction with model SMB_U_T_04_11_91, which is given in Equation 3.10.

$$u_t = (0.624)\, u_{t-1} - (0.380)\, u_{t-365} + (0.225)\, u_{t-366} + e_t \qquad (3.10)$$

The residuals produced by this model were found to be uncorrelated. The mean of the residuals was close to zero and the histogram of the residual frequencies was approximately normally distributed. There were no outliers and there was no apparent trend. The Box-Pierce method and the Portmanteau lack-of-fit test also indicated the adequacy of the above model.

The one day forecasts produced by model SMB_U_T_04_11_91 were very good, with a RMSE of 10.5 EC units and an AAPE of 1.1% for 1991. However, the 14 day forecasts were very noisy, with a RMSE of 109.0 EC units and an AAPE of 11.8% for

1991. The 14 day forecasts were able to be improved significantly (RMSE = 93.0 EC units, AAPE = 10.2%) by using a one-sided filter.

3.7 Development of Multivariate Time Series Model

3.7.1 Introduction

In this section, the development of the multivariate time series (VARIMA) model for the predicition of salinity in the River Murray at Murray Bridge is described. The objective of this section is to develop a VARIMA model, which provides a basis of comparison for the univariate time series (ARIMA) and the multivariate ANN models.

The steps that were followed to develop the VARIMA model are outlined in Figure 3.146. Forecasting periods of 1, 5 and 14 days were used, as was the case with the ARIMA and ANN models. For all models, data from 1987 to 1990 were used for parameter estimation and data from 1991 were used for testing in order to obtain real time forecasts, as was the case when the ARIMA and ANN models were developed.

3.7.2 Choice of Component Time Series

All available time series (Table 3.2) were considered, with the exception of flow at Overland Corner, as it is almost identical to Flow at Lock 1 Lower.

3.7.3 Inspection of Plots of the Time Series

Plots and discussions of the component time series are given in Section 3.3.3.1. At this stage of modelling, it is assumed that the distributions of the component time series approach normality. At the diagnostic checking stage, the model residuals will be examined to check whether the model is adequate. If this is found to be the case, there is no need to transform the original data using a Box-Cox transform (Verbyla, 1995).

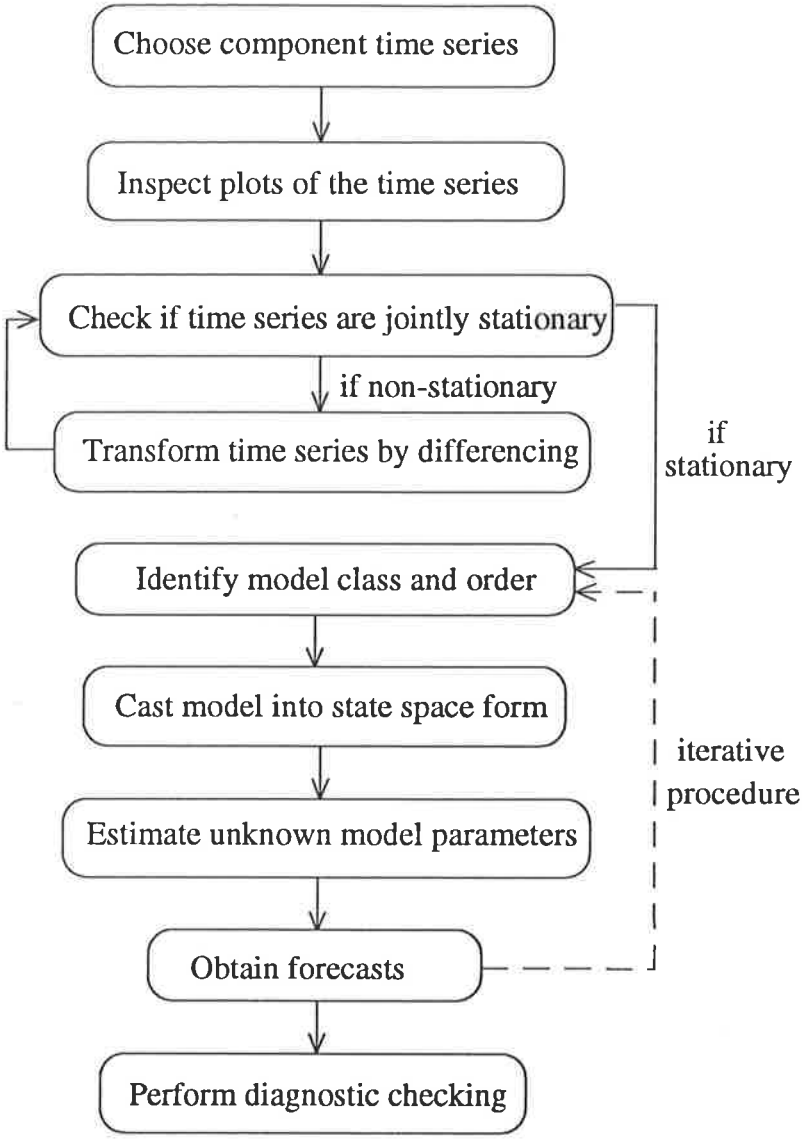


Figure 3.146: Steps in the Development of the VARIMA Model

3.7.4 Checking of the Time Series for Joint Stationarity

The ACF and the PACF were obtained for each component time series using computer program 'APCORR'. The maximum lag used was 1095, for reasons explained in Section 3.4.2. The results obtained are summarised in Table 3.103.

Each of the component time series was found to be non-stationary, as indicated by ACFs that die down extremely slowly. Consequently, each original time series needed to be transformed into a stationary time series.

Table 3.103: Behaviour of the ACF and PACF of the Component Time Series

Time Series	ACF	PACF
SMN	Dies down extremely slowly	Cuts off at a lag of 717
SMO	Dies down extremely slowly	Cuts off at a lag of 686
SWE	Dies down extremely slowly	Cuts off at a lag of 426
SLO	Dies down extremely slowly	Cuts off at a lag of 610
SL5U	Dies down extremely slowly	Cuts off at a lag of 553
FL1L	Dies down extremely slowly	Cuts off at a lag of 409
LMB	Dies down extremely slowly	Cuts off at a lag of 515
LMN	Dies down extremely slowly	Cuts off at a lag of 676
LL1L	Dies down extremely slowly	Cuts off at a lag of 741
LL1U	Dies down extremely slowly	Cuts off at a lag of 825
LMO	Dies down extremely slowly	Cuts off at a lag of 410
LWE	Dies down extremely slowly	Cuts off at a lag of 452
LOC	Dies down extremely slowly	Cuts off at a lag of 415
LLO	Dies down extremely slowly	Cuts off at a lag of 129

3.7.5 Transformation of the Time Series by Differencing

Each of the component time series was differenced in accordance with Equation 2.56. As suggested by the development of the univariate time series (ARIMA) model (Section 3.6), the degree of non-seasonal differencing (d) and the degree of seasonal differencing (D) were each chosen to be one. The seasonality (L) was chosen to be 365 days. It should be noted that each time series was differenced the same number of times in order to maintain the phase relationship between them.

Differencing was carried out using computer program 'DIFFER'. The following criteria were adopted for joint stationarity (see Section 2.2.13.4):

- The ACF and the PACF of each time series have to cut off at a lag less than or equal to 2L+2 (i.e. 732 days).
- The CCF between each pair of time series has to cut off at a lag less than or equal to 2L+2 (i.e. 732 days).

The ACF and the PACF were obtained for each differenced time series using computer program 'APCORR'. The maximum lag used was 1095, for reasons discussed in Section 3.4.3. The ACF and the PACF were examined for significant (non-zero) values in accordance with Equations 2.40 and 2.43 respectively. The lags at which significant

values occurred in the ACF and the PACF are shown in Tables 3.104 to 3.106. The numbers in brackets indicate the total number of significant values.

Table 3.104: Lags at which Values of the ACF and PACF are Significantly Different from Zero for the Differenced Salinity Time Series

Time Series	Lags at which significant values occur (days)	
	ACF	PACF
SMN	1, 2, 3, 4, 9, 15, 16, 19, 27, 202, 203, 204, 205, 338, 363, 364, 365, 366 (18)	1, 3, 9, 19, 20, 27, 28, 33, 40, 63, 65, 84, 87, 132, 157,183, 202, 203, 230, 248, 256, 309, 318, 326, 329, 338, 346, 355, 364, 365, 366, 730, 731 (33)
SMO	1, 19, 136, 166, 199, 201, 229, 364, 365 (9)	1, 19, 34, 45, 54, 36, 156, 166, 190, 191, 199, 200, 210, 221, 224, 305, 312, 365, 366, 500, 730 (21)
SWE	1, 2, 3, 8, 9, 10, 11, 12, 177, 363, 364, 365,366, 367, 426, 427 (16)	1, 2, 8, 23, 38, 50, 56, 70, 76, 90, 100, 107, 133, 150, 169, 175, 197, 206, 261, 275, 287, 297, 306, 315, 362, 363, 364, 365, 366, 548 (30)
SLO	1, 2, 9, 10, 11, 20, 21, 31, 34, 58, 59, 124, 168, 364, 365, 366 (16)	1, 2, 3, 4, 5, 8, 18, 20, 25, 58, 61, 86, 128, 153, 169, 178, 190, 196, 200, 204, 297, 324, 328, 342, 362, 364, 365, 366, 367, 422, 456, 583, 729 (33)
SL5U	1, 2, 11, 40, 41, 325, 364, 365, 366 (9)	1, 2, 11, 12, 14, 19, 22, 39, 56, 60, 64, 86, 106, 164, 177, 232, 290, 308, 324, 364, 365, 366, 378, 730 (24)

Table 3.105: Lags at which Values of the ACF and PACF are Significantly Different from Zero for the Differenced Level Time Series

Time Series	Lags at which significant values occur (days)	
	ACF	PACF
LMB	1, 2, 3, 4, 14, 29, 30, 32, 33, 42, 57, 61, 62, 68, 70, 77, 87, 132, 159, 238, 363, 365 (22)	1, 2, 3, 4, 14, 16, 29, 43, 68, 77, 79, 113, 132, 136, 159, 209, 238, 275, 290, 323, 362, 363, 365, 367, 730 (25)
LMN	1, 2, 3, 6, 9, 10, 15, 29, 32, 33, 50, 61, 62, 332, 365 (15)	1, 2, 3,9, 14, 18, 29, 33, 79, 176, 268, 314, 315, 330, 341, 365, 729, 730 (18)
LL1L	1, 2, 3, 4, 5, 6, 7, 15, 25, 82, 133, 365, 386, 388, 389, 391 (16)	1, 4, 5, 6, 15, 28, 29, 34, 47, 50, 77, 82, 113, 119, 120, 133, 134, 149, 183, 249, 258, 300, 312, 343, 349, 365, 366, 376, 571 (29)
LL1U	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 25, 40, 67, 75, 77, 106, 364, 365, 366, 389, 390 (22)	1, 3, 4, 6, 7, 17, 31, 36, 39, 46, 54, 56, 66, 77, 156, 220, 221, 259, 294, 314,364, 365, 366, 368, 474, 730 (26)
LMO	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 18, 19, 24, 25, 26, 363, 364, 365, 366, 367, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393 (31)	1, 2, 3, 9, 16, 17, 29, 31, 43, 45, 48, 74, 113, 134, 155, 156, 177, 191, 207, 279, 360, 362, 363, 364, 365, 366 (26)
LWE	1, 2, 3, 4, 5, 6, 7, 8, 9, 22, 23, 363, 364, 365, 366, 385, 386, 387, 388, 389, 390, 391, 392 (23)	1, 2, 3, 4, 21, 30, 35, 36, 41, 44, 56, 57, 64, 85, 88, 90, 146, 183, 217, 275, 301, 310, 329, 338, 362, 363, 364, 365, 366, 369, 450, 729, 731 (33)
LOC	1, 2, 3, 4, 5, 6, 7, 8, 9, 18, 19, 20, 21, 22, 364, 365, 366, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392 (27)	1, 2, 3, 5, 14, 17, 74, 144, 220, 224, 238, 333, 362, 363, 364, 365, 366, 431, 729, 731 (20)
LLO	1, 2, 3, 4, 5, 6, 7, 8, 17, 18, 19, 20, 21, 22, 23, 164, 165, 166, 201, 364, 365, 366, 367, 382, 383, 384, 385, 386, 387, 388 (30)	1, 2, 3, 5, 6, 8, 13, 16, 18, 23, 31, 143, 162, 176, 177, 197, 228, 307, 332, 361, 362, 363, 364, 366, 371 (25)

Table 3.106: Lags at which Values of the ACF and PACF are Significantly Different from Zero for the Differenced Flow Time Series

Time Series	Lags at which significant values occur (days)	
	ACF	PACF
FL1L	1, 2, 3, 4, 5, 6, 7, 8, 9, 364, 365, 366, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394 (24)	1, 3, 17, 32, 34, 73, 172, 175, 191, 194, 197, 208, 220, 225, 253, 307, 321, 360, 364, 365, 366, 415, 731 (23)

As all ACFs and PACFs cut off at a lag less than or equal to 2L+2, all differenced time series may be considered stationary.

The plots of the ACF and PACF for each differenced time series (not shown here), indicate that the values at low lags (1 to 5) and at one seasonal lag (364 to 367) are considerably greater than the bounds of significance. All other "significant" values are close to the bounds of significance and may hence be ignored, as explained in Section 3.6.6.

The CCFs between all differenced component time series were obtained using computer program 'CROSSCORR'. The maximum lag used was 1095. The CCFs (not shown here) were examined for significant (non-zero) values. Values were considered to be significant if they exceeded the value given by Equation 2.41.

All CCFs cut off at a lag less than or equal to one seasonal lag. Consequently, the set of differenced time series were considered covariance stationary, as the ACF and the PACF for each differenced time series cut off at a lag less than or equal to 2L+2, and the CCF between each pair of differenced time series cut off at a lag less than or equal to 2L+2.

3.7.6 Identification of the Class and Initial Order of the Model Predicting Salinity at Murray Bridge

3.7.6.1 Model Class

The relationship between salinity at a particular location and salinities, flows and river levels at upstream sites is unidirectional (i.e. the salinity at a particular location is affected by upstream salinities, flows and river levels, but the reverse does not hold). As a result, the parameter matrices in the VARIMA model are upper (or lower) triangular, resulting in a transfer function (TFN) model (Section 2.2.13.2).

3.7.6.2 Model Order

An analytical approach, based on the method proposed by Haugh and Box (1977) (Section 2.2.13.5), was used to determine the initial order of the AR and MA parameters. The method involves pre-whitening each of the component time series by fitting a univariate time series model to each component time series and carrying out a cross-correlation analysis on the residuals obtained.

Fitting of ARIMA models to the component time series

The pre-whitened residuals were obtained by fitting ARIMA models to each of the differenced component time series, using the Box-Jenkins methodology summarised in Section 2.2.12. As shown in Section 3.7.5, each of the component time series was stationary after being differenced in accordance with Equation 2.56, using d = 1, D = 1 and L = 365.

As observed in Section 3.7.5, the peaks in the ACFs and PACFs that are markedly greater than the bounds of significance occur at lags 1 to 5 and at lags 364 to 367. This applies to all differenced time series. Consequently, the initial model developed for each differenced time series contained a total of 18 parameters, as set out in Table 3.107.

Table 3.107: Initial Choice of Parameter Types and Corresponding Lags for each ARIMA Model

Parameter Type	Lag	Parameter Type	Lag
AR	1	MA	1
AR	2	MA	2
AR	3	MA	3
AR	4	MA	4
AR	5	MA	5
AR	364	MA	364
AR	365	MA	365
AR	366	MA	366
AR	367	MA	367

The initial estimates for all AR parameters (ϕ_i) and all MA parameters (θ_i) were chosen to be 0.1 in accordance with the recommendations by Bowermann and O'Connell (1979). The following procedure was followed in order to finalise model dimensions, and to obtain final parameter estimates:

1. Estimate the model parameters for the initial model using computer program 'PAREST'.
2. Reduce the size of the model by eliminating any parameters that are not significant (i.e. parameters that are smaller than $2/\sqrt{n}$).
3. Use computer program 'PAREST' to obtain the final parameter estimates for the revised model.

It should be noted that the aim of this procedure was not to obtain the best possible model for the data set available, but to obtain a set of uncorrelated residuals. It was therefore unnecessary to refine the model any further, despite the fact that some of the models contain parameters that are not significant. The model parameters obtained using the above procedure are summarised in Tables 3.108 to 3.121.

Table 3.108: Salinity at Mannum - Model Parameters (Model SMN_U_T_01_11_91)

AR	Order	MA	Order
+0.3471	1	-0.2814	1
+0.3979	2	+0.3170	2
-0.1785	3	-0.0665	3
-0.5002	365	-0.0484	4
+0.1041	366	+0.1149	363
+0.1610	367	+0.2214	364
		-0.0013	367

Table 3.109: Salinity at Morgan - Model Parameters (Model SMO_U_T_01_11_91)

AR	Order	MA	Order
+0.6474	1	+0.3653	1
+0.0394	2	+0.2316	2
-0.3817	365	-0.0300	364
+0.2092	366	+0.4531	365
		-0.2743	366
		-0.0783	367

Table 3.110: Salinity at Waikerie - Model Parameters (Model SWE_U_T_01_11_91)

AR	Order	MA	Order
+0.2978	1	-0.5586	1
+0.1928	2	-0.1290	2
-0.0751	3	-0.0668	3
-0.4133	365	+0.1798	364
+0.0959	367	+0.2330	365

Table 3.111: Salinity at Loxton - Model Parameters (Model SLO_U_T_01_11_91)

AR	Order	MA	Order
+0.7423	1	-0.2924	1
-0.3152	2		
+0.0734	3		
-0.0648	4		
+0.0345	5		
-0.4856	365		
+0.3127	366		
-0.1336	367		

Table 3.112: Salinity at Lock 5 Upper - Model Parameters (Model SL5U_U_T_01_11_91)

AR	Order	MA	Order
-0.0381	1	-0.5952	1
+0.2200	2	+0.1036	3
+0.0693	3		
-0.5084	365		
-0.0275	366		
+0.0847	367		

Table 3.113: Flow at Lock 1 Lower - Model Parameters (Model FL1L_U_T_01_11_91)

AR	Order	MA	Order
+0.3870	1	-0.5293	1
+0.3541	2	+0.2923	365
-0.4255	365	+0.1381	366
+0.1511	366		
+0.2284	367		

Table 3.114: Level at Murray Bridge - Model Parameters (Model LMB_U_T_01_11_91)

AR	Order	MA	Order
-0.1513	1	-0.4538	1
-0.4519	365	+0.2095	365
-0.1289	366		
-0.1082	367		

Table 3.115: Level at Mannum - Model Parameters (Model LMN_U_T_01_11_91)

AR	Order	MA	Order
+0.3708	1	+0.1473	364
-0.1745	2	+0.2135	365
-0.4085	365	-0.0811	366
+0.0548	366	-0.0827	367
-0.0955	367		

Table 3.116: Level at Lock 1 Lower - Model Parameters (Model LL1L_U_T_01_11_91)

AR	Order	MA	Order
+0.9117	1	+0.4409	1
+0.0492	2	+0.2974	2
-0.3315	365	+0.3241	365
+0.3144	366	-0.1738	366
+0.0546	367		

Table 3.117: Level at Lock 1 Upper - Model Parameters (Model LL1U_U_T_01_11_91)

AR	Order	MA	Order
+0.6561	1	+0.0720	1
+0.3600	2	+0.5591	2
-0.0875	3	+0.0840	3
+0.1231	4	+0.0649	4
-0.1084	5	-0.0539	5
+0.0473	364	+0.1524	364
-0.4979	365	+0.2825	365
+0.30566	366	-0.01836	366
+0.20166	367	-0.08420	367

Table 3.118: Level at Morgan - Model Parameters (Model LMO_U_T_01_11_91)

AR	Order	MA	Order
+0.9304	1	+0.1220	1
+0.3825	2	+0.5943	2
-0.3280	3	+0.0375	3
+0.1596	4	+0.1321	4
-0.1631	5	+0.0313	5
+0.0520	364	+0.1253	364
-0.4517	365	+0.4081	365
+0.3732	366	-0.0977	366
+0.0452	367	-0.4040	367

Table 3.119: Level at Waikerie - Model Parameters (Model LWE_U_T_01_11_91)

AR	Order	MA	Order
+0.6023	1	-0.3205	1
+0.3240	2	+0.2747	2
-0.5008	365	+0.3087	3
+0.3179	366	+0.1371	4
+0.2566	367	+0.0730	364
		+0.2779	365
		+0.2145	366

Table 3.120: Level at Overland Corner - Model Parameters (Model LOC_U_T_01_11_91)

AR	Order	MA	Order
+1.391	1	+0.4371	1
-0.4083	2	+0.2984	2
-0.3980	365	+0.0899	4
+0.5881	366	-0.0094	5
-0.1727	367	+0.39440	365
		-0.0821	366
		-0.1758	367

Table 3.121: Level at Loxton - Model Parameters (Model LLO_U_T_01_11_91)

AR	Order	MA	Order
+1.102	1	+0.2250	2
-0.1252	2	+0.1544	3
-0.4241	365	+0.1820	4
+0.4321	366	+0.1763	5
+0.0147	367	+0.3193	365
		-0.0849	367

Diagnostic checking was carried out for each of the above models. The residuals were obtained using computer program 'PAREST' and were found to have zero mean, no outliers and no apparent trend. The variances of the residuals were found to be constant, and the histogram of residual frequencies (not shown here) was normally distributed, indicating that there is no need to transform the original data using a Box-Cox transform (Hipel, 1985).

The Box-Pierce method was then used to examine the residuals for each model. The ACFs and the PACFs of the residuals (not shown here) were obtained with the aid of computer program 'APCORR'. None of the values in the ACFs or the PACFs were significant at low lags (i.e. 1 to 8) or at seasonal lags (i.e. 360 to 370). Any values that were significant were only marginally greater than the bounds of significance. Consequently, the Box-Pierce method also suggests that the residuals are uncorrelated.

Finally, the Portmanteau lack-of-fit test was used to check the whiteness of the residuals for each model. The Box-Pierce chi-square statistics (Q^{BP}) were calculated using computer program 'APCORR'. The value of k' used was 365, as suggested in Section

3.6.7. $\chi^2_{(5)}$ was calculated in accordance with Equation 3.6. The values of Q^{BP} and $\chi^2_{(5)}$ for the various models are shown in Table 3.122.

Table 3.122: Values of Q^{BP} and $\chi^2_{(5)}$ for the Univariate Models Considered

Model	m'	Q^{BP}	$\chi^2_{(5)}$
SMN_U_T_01_11_91	352	245	396
SMO_U_T_01_11_91	355	227	400
SWE_U_T_01_11_91	355	306	400
SLO_U_T_01_11_91	356	270	401
SL5U_U_T_01_11_91	357	249	402
FL1L_U_T_01_11_91	357	245	402
LMB_U_T_01_11_91	359	323	404
LMN_U_T_01_11_91	356	251	401
LL1L_U_T_01_11_91	356	268	401
LL1U_U_T_01_11_91	347	239	391
LMO_U_T_01_11_91	347	263	391
LOC_U_T_01_11_91	353	286	397
LWE_U_T_01_11_91	353	231	397
LLO_U_T_01_11_91	354	283	399

Table 3.122 shows that each Box-Pierce statistic is smaller than its corresponding value of $\chi^2_{(5)}$. It can therefore be concluded that the residuals obtained from each model are uncorrelated.

Calculation of cross-correlations

Once it was established that the residuals have the characteristics of white noise, the CCFs between the residuals of model SMB_U_T_04_11_91 and the residuals of the ARIMA models for the other component time series were obtained using computer program 'CROSSCORR'. The maximum lag used was 400. The lags at which maximum correspondence occurs, and the corresponding cross-correlation values, are shown in Table 3.123.

Table 3.123: Significant Cross-Correlations Between the Residuals of Salinity at Murray Bridge and the Residuals of the Other Component Time Series

Time Series:	Lag (days)														
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
SMN												0.14	0.26	0.18	0.18
SMO									0.2	0.15	0.1				
SWE					0.16	0.22	0.1								
SLO			0.14			0.14	0.09								
SL5U	0.16				0.11										
FL1L						-0.11									
LMB															
LMN															
LL1L							-0.09			-0.1		-0.09			
LL1U					-0.18						-0.1				
LMO															
LWE					-0.09										
LOC															
LLO		-0.09				-0.09									

It can be seen that:

- As would be expected, lags at which maximum correspondence occurs are greater for sites located further upstream.
- Salinity values are more highly correlated than the flow or level data.
- There are no significant cross-correlations between salinity at Murray Bridge and levels at Murray Bridge, Mannum, Morgan and Overland Corner.

Examination of the relationship between the level data at various locations

In order to reduce the size of the multivariate model, the relationships between the level data at the various locations were examined. The relationships between the flow data at Lock 1 Lower and the river level data at the various locations were also examined. By inspection of the various plots of the level and flow data (Figures 3.42 to 3.51), it can be seen that the plots are almost identical, apart from a difference in base flow. In order to assess the strength of the relationship between the various time series, the CCF was calculated between the series of residuals with the aid of computer program 'CROSSCORR'. The maximum cross-correlations between the various series of residuals are shown in Table 3.124. It can be seen that:

- There is a strong relationship between the level at Murray Bridge and the level at Mannum. These time series are not strongly related to the other time series examined.
- The levels at Lock 1 Lower, Lock 1 Upper, Morgan, Waikerie, Overland Corner and Loxton, as well as the flow at Lock 1 Lower, are strongly related.

As the levels at Murray Bridge and at Mannum did not seem to be related to the salinity at Murray Bridge, they were discarded from further consideration.

By considering the plots of the time series and the maximum cross-correlations obtained, the levels at Lock 1 Lower, Morgan, Waikerie, Overland Corner and Loxton, as well as the flow at Lock 1 Lower, may be deemed to be dependent variables. Consequently, it was decided to use only one of these variables in the multivariate model. The level at Loxton was chosen, as it is highly correlated to all other variables. In addition, Loxton is the most upstream location, and hence values at greater lags are related to the salinity at Murray Bridge. This is desirable for longer term forecasts.

Figure 3.47 shows that the plot of level at Lock 1 Lower looks significantly different than those of the other level data (Figures 3.46 and 3.48 to 3.51) and that of flow at Lock 1 Lower (Figure 3.42). This can be expected, as the water level at Lock 1 Upper

is controlled by Lock operation. However, there is quite a strong correlation between the level at Lock 1 Upper and salinity at Murray Bridge, as shown in Table 3.123. Consequently, the level at Lock 1 Upper was also chosen as an input variable for the multivariate time series model.

Table 3.124: Maximum Cross-Correlations Between Residuals of Level and Flow Data

Time Series:	FL1L	LMB	LMN	LL1L	LL1U	LMO	LWE	LOC	LLO
FL1L		0.12	0.15	0.46	0.47	0.38	0.35	0.36	0.27
LMB	0.12		0.57	0.19	0	0	0	0	0
LMN	0.15	0.57		0.1	0.14	0.07	0	0	0
LL1L	0.46	0.19	0.1		0.23	0.27	0.32	0.23	0.21
LL1U	0.47	0	0.14	0.23		0.31	0.31	0.28	0.21
LMO	0.38	0	0.07	0.27	0.31		0.37	0.3	0.23
LWE	0.36	0	0	0.23	0.28	0.3		0.48	0.27
LOC	0.35	0	0	0.32	0.31	0.37	0.48		0.35
LLO	0.27	0	0	0.21	0.21	0.23	0.35	0.27	

By considering the CCFs between all possible component time series, the time series chosen to provide the input variables for the VARIMA model for the prediction of salinity at Murray Bridge include:

1. Salinity at Murray Bridge (SMB).
2. Salinity at Mannum (SMN).
3. Salinity at Morgan (SMO).
4. Salinity at Waikerie (SWE).
5. Salinity at Loxton (SLO).
6. Salinity at Lock 5 Upper (SL5U).
7. Level at Lock 1 Upper (LL1U).
8. Level at Loxton (LLO).

Choice of the order of the AR and MA parameters

Having identified which time series to use as inputs to the multivariate model, the initial orders of the AR parameters were obtained with the aid of Table 3.123. Choosing the order of the MA parameters was more difficult. The values of the MA parameters obtained as part of the development of the univariate ARIMA models for the prediction of salinity at Murray Bridge (Section 3.6) were used as a guide. The initial order of the

AR and MA parameters used to predict salinity at Murray Bridge (model SMB_M_T_11_11_91) are listed in Table 3.125. The total number of AR parameters is 22 and the total number of MA parameters is 3.

Table 3.125: Order of MA and AR Parameters for the Initial Multivariate Model of Salinity at Murray Bridge (Model SMB_M_T_11_11_91)

AR Parameters		MA Parameters
Time Series	Order	Order
SMB	1, 365, 366	365, 366, 367
SMN	1, 2, 3, 4	
SMO	5, 6, 7	
SWE	9, 10, 11	
SLO	9, 10, 13	
SL5U	11, 15	
LL1U	5, 11	
LLO	10, 14	

3.7.7 Identification of the Initial Order of the Models for the Time Series Required for the Prediction of Salinity at Murray Bridge

In order to obtain the best possible multi-step forecasts of salinity at Murray Bridge, it is also necessary to obtain forecasts of the other component time series. However, as the focus in this research is on forecasting salinity at Murray Bridge, it was decided to restrict the range of possible input time series for the models predicting these variables to the time series identified as being strongly related to salinity at Murray Bridge (i.e. SMN, SMO, SWE, SLO, SL5U, LL1U and LLO are predicted using inputs from SMN, SMO, SWE, SLO, SL5U, LL1U and LLO only).

The CCFs between the residual time series of the variable to be predicted and the residual time series of the upstream variables (Tables 3.126 to 3.130) were used in conjunction with the univariate time series models developed in Section 3.7.6 to identify the initial order of the AR and MA model parameters. It should be noted that there is no table showing the cross-correlations between the salinity at Lock 5 Upper and its upstream variables, as none of the cross-correlations were significant.

Table 3.126: Significant Cross-Correlations Between the Residuals of Salinity at Mannum and the Residuals of the Time Series Upstream of Mannum

Time Series	Lag (days)											
	14	13	12	11	10	9	8	7	6	5	4	3
SMO								0.13	0.16	0.21	0.15	0.09
SWE					0.15	0.21	0.25					
SLO	0.14			0.13								
SL5U	0.14	0.13										
LL1U							-0.1					
LLO			-0.13									

Table 3.127: Significant Cross-Correlations Between the Residuals of Salinity at Morgan and the Residuals of the Time Series Upstream of Morgan

Time Series	Lag (days)								
	9	8	7	6	5	4	3	2	1
SWE				0.2	0.24	0.23	0.18		
SLO	0.13	0.12		0.13					
SL5U	0.18		0.09						
LLO	-0.1	-0.1	-0.1	-0.1					

Table 3.128: Significant Cross-Correlations Between the Residuals of Salinity at Waikerie and the Residuals of the Time Series Upstream of Waikerie

Time Series	Lag (days)								
	9	8	7	6	5	4	3	2	1
SLO					0.11	0.14	0.14	0.13	0.09
SL5U				0.18	0.12	0.1	0.11		
LLO				-0.1	-0.1	-0.2	-0.2		

Table 3.129: Significant Cross-Correlations Between the Residuals of Salinity at Loxton and the Residuals of the Time Series Upstream of Loxton

Time Series	Lag (days)								
	9	8	7	6	5	4	3	2	1
SL5U					0.11	0.1	0.26	0.23	
LLO					-0.1				-0.1

Table 3.130: Significant Cross-Correlations Between the Residuals of Level at Lock 1 Upper and the Residuals of the Time Series Upstream of Lock 1

Time Series:	Lag (days)								
	9	8	7	6	5	4	3	2	1
LLO				0.2	0.1	0.21			

The initial order of the AR and MA parameters used to predict the various differenced time series are given in Tables 3.131 to 3.137.

Table 3.131: Order of MA and AR Parameters for the Initial Multivariate Model Predicting Salinity at Mannum (Model SMN_M_T_02_11_91)

AR Parameters		MA Parameters
Time Series	Order	Order
SMN	1, 2, 3, 365, 366, 367	363, 364, 367 (3 Total)
SMO	3, 4, 5, 6, 7	
SWE	8, 9, 10	
SLO	11, 12, 13, 14	
SL5U	13, 14	
LL1U	8	
LLO	12 (22 Total)	

Table 3.132: Order of MA and AR Parameters for the Initial Multivariate Model Predicting Salinity at Morgan (Model SMO_M_T_02_11_91)

AR Parameters		MA Parameters
Time Series	Order	Order
SMO	1, 2, 365, 366	364, 365, 366, 367 (4 Total)
SWE	3, 4, 5, 6	
SLO	6, 7, 8, 9	
SL5U	7, 8, 9	
LLO	6, 7, 8, 9 (19 Total)	

Table 3.133: Order of MA and AR Parameters for the Initial Multivariate Model Predicting Salinity at Waikerie (Model SWE_M_T_02_11_91)

AR Parameters		MA Parameters
Time Series	Order	Order
SWE	1, 2, 3, 365, 367	364, 365 (2 Total)
SLO	1, 2, 3, 4, 5	
SL5U	3, 4, 5, 6	
LLO	3, 4, 5, 6 (18 Total)	

Table 3.134: Order of MA and AR Parameters for the Initial Multivariate Model Predicting Salinity at Loxton (Model SLO_M_T_02_11_91)

AR Parameters		MA Parameters
Time Series	Order	Order
SLO	1, 2, 3, 4, 5, 365, 366, 367	364, 365, 366, 367 (4 Total)
SL5U	2, 3, 4, 5	
LLO	1, 2, 3, 4, 5 (17 Total)	

Table 3.135: Order of MA and AR Parameters for the Initial Multivariate Model Predicting Salinity at Lock 5 Upper (Model SL5U_M_T_02_11_91)

AR Parameters		MA Parameters
Time Series	Order	Order
SL5U	1, 2, 3, 365, 366, 367 (6 Total)	364, 365, 366, 367 (4 Total)

Table 3.136: Order of MA and AR Parameters for the Initial Multivariate Model Predicting Level at Lock 1 Upper (Model LL1U_M_T_02_11_91)

AR Parameters		MA Parameters
Time Series	Order	Order
LL1U	1, 2, 3, 4, 5, 364, 365, 366, 367	364, 365, 366, 367 (4 Total)
LLO	4, 5, 6 (12 Total)	

Table 3.137: Order of MA and AR Parameters for the Initial Multivariate Model Predicting Level at Loxton (Model LLO_U_T_02_11_91)

AR Parameters		MA Parameters
Time Series	Order	Order
LLO	1, 2, 3, 364, 365, 366, 367 (7 Total)	364, 365, 366, 367 (4 Total)

3.7.8 Formulation of the Model in State Space Form

The VARIMA model for the prediction of salinity at Murray Bridge was formulated in state space form for ease of computation. The general form of the state space model adopted is shown below. It should be noted that the dimension of the problem is very large (the dimension of $\mathbf{Z}_t$ is 5864 x 1), as it deals with daily data that has seasonal variation.

Transition Equation:

$$\mathbf{Z}_t = \Phi\, \mathbf{Z}_{t-1} + \mathbf{E}_t$$

$$\begin{bmatrix} z_{SMB,t} \\ z_{SMN,t} \\ z_{SMO,t} \\ z_{SWE,t} \\ z_{SLO,t} \\ z_{SL5U,t} \\ z_{LL1U,t} \\ z_{LLO,t} \\ e_{SMB,t} \\ e_{SMN,t} \\ e_{SMO,t} \\ e_{SWE,t} \\ e_{SLO,t} \\ e_{SL5U,t} \\ e_{LL1U,t} \\ e_{LLO,t} \end{bmatrix} = \left[\begin{array}{cccccc|cccccc} \Phi_{11} & \Phi_{12} & \Phi_{13} & \cdots & \Phi_{17} & \Phi_{18} & \Theta_{11} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \Phi_{22} & \Phi_{23} & \cdots & \Phi_{27} & \Phi_{28} & 0 & \Theta_{22} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \Phi_{33} & \cdots & \Phi_{37} & \Phi_{38} & 0 & 0 & \Theta_{33} & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot & \cdot & & & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot & \cdot & & & & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 0 & \Phi_{88} & 0 & 0 & 0 & \cdots & 0 & \Theta_{88} \\ \hline 0 & 0 & 0 & \cdots & 0 & 0 & \Theta_I & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \Theta_I & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & & & \cdot & \cdot & \cdot & \cdot & & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & & & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot & \cdot & & & & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot & \cdot & & & & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & \Theta_I \end{array}\right] \begin{bmatrix} z_{SMB,t-1} \\ z_{SMN,t-1} \\ z_{SMO,t-1} \\ z_{SWE,t-1} \\ z_{SLO,t-1} \\ z_{SL5U,t-1} \\ z_{LL1U,t-1} \\ z_{LLO,t-1} \\ e_{SMB,t-1} \\ e_{SMN,t-1} \\ e_{SMO,t-1} \\ e_{SWE,t-1} \\ e_{SLO,t-1} \\ e_{SL5U,t-1} \\ e_{LL1U,t-1} \\ e_{LLO,t-1} \end{bmatrix} + \begin{bmatrix} E_{SMB,t} \\ E_{SMN,t} \\ E_{SMO,t} \\ E_{SWE,t} \\ E_{SLO,t} \\ E_{SL5U,t} \\ E_{LL1U,t} \\ E_{LLO,t} \\ E_{SMB,t} \\ E_{SMN,t} \\ E_{SMO,t} \\ E_{SWE,t} \\ E_{SLO,t} \\ E_{SL5U,t} \\ E_{LL1U,t} \\ E_{LLO,t} \end{bmatrix}$$

where:

$$\mathbf{z}_{SMB,t} = \begin{bmatrix} SMB_t \\ SMB_{t-1} \\ SMB_{t-2} \\ . \\ . \\ . \\ . \\ SMB_{t-365} \\ SMB_{t-366} \\ SMB_{t-367} \end{bmatrix}, \quad \mathbf{z}_{SMN,t} = \begin{bmatrix} SMN_t \\ SMN_{t-1} \\ SMN_{t-2} \\ . \\ . \\ . \\ . \\ SMN_{t-365} \\ SMN_{t-366} \\ SMN_{t-367} \end{bmatrix}, \quad \mathbf{z}_{SMO,t} = \begin{bmatrix} SMO_t \\ SMO_{t-1} \\ SMO_{t-2} \\ . \\ . \\ . \\ . \\ SMO_{t-365} \\ SMO_{t-366} \\ SMO_{t-367} \end{bmatrix}, \quad \mathbf{z}_{SWE,t} = \begin{bmatrix} SWE_t \\ SWE_{t-1} \\ SWE_{t-2} \\ . \\ . \\ . \\ . \\ SWE_{t-365} \\ SWE_{t-366} \\ SWE_{t-367} \end{bmatrix},$$

$$\mathbf{z}_{SLO,t} = \begin{bmatrix} SLO_t \\ SLO_{t-1} \\ SLO_{t-2} \\ . \\ . \\ . \\ . \\ SLO_{t-365} \\ SLO_{t-366} \\ SLO_{t-367} \end{bmatrix}, \quad \mathbf{z}_{SL5U,t} = \begin{bmatrix} SL5U_t \\ SL5U_{t-1} \\ SL5U_{t-2} \\ . \\ . \\ . \\ . \\ SL5U_{t-365} \\ SL5U_{t-366} \\ SL5U_{t-367} \end{bmatrix}, \quad \mathbf{z}_{LL1U,t} = \begin{bmatrix} LL1U_t \\ LL1U_{t-1} \\ LL1U_{t-2} \\ . \\ . \\ . \\ . \\ LL1U_{t-365} \\ LL1U_{t-366} \\ LL1U_{t-367} \end{bmatrix}, \quad \mathbf{z}_{LLO,t} = \begin{bmatrix} LLO_t \\ LLO_{t-1} \\ LLO_{t-2} \\ . \\ . \\ . \\ . \\ LLO_{t-365} \\ LLO_{t-366} \\ LLO_{t-367} \end{bmatrix},$$

$$\mathbf{e}_{SMB,t} = \begin{bmatrix} e_{SMB,t} \\ e_{SMB,t-1} \\ e_{SMB,t-2} \\ . \\ . \\ . \\ . \\ e_{SMB,t-365} \\ e_{SMB,t-366} \end{bmatrix}, \quad \mathbf{e}_{SMN,t} = \begin{bmatrix} e_{SMN,t} \\ e_{SMN,t-1} \\ e_{SMN,t-2} \\ . \\ . \\ . \\ . \\ e_{SMN,t-365} \\ e_{SMN,t-366} \end{bmatrix}, \quad \mathbf{e}_{SMO,t} = \begin{bmatrix} e_{SMO,t} \\ e_{SMO,t-1} \\ e_{SMO,t-2} \\ . \\ . \\ . \\ . \\ e_{SMO,t-365} \\ e_{SMO,t-366} \end{bmatrix}, \quad \mathbf{e}_{SWE,t} = \begin{bmatrix} e_{SWE,t} \\ e_{SWE,t-1} \\ e_{SWE,t-2} \\ . \\ . \\ . \\ . \\ e_{SWE,t-365} \\ e_{SWE,t-366} \end{bmatrix},$$

$$\mathbf{e}_{SLO,t} = \begin{bmatrix} e_{SLO,t} \\ e_{SLO,t-1} \\ e_{SLO,t-2} \\ . \\ . \\ . \\ . \\ . \\ e_{SLO,t-365} \\ e_{SLO,t-366} \end{bmatrix}, \quad \mathbf{e}_{SL5U,t} = \begin{bmatrix} e_{SL5U,t} \\ e_{SL5U,t-1} \\ e_{SL5U,t-2} \\ . \\ . \\ . \\ . \\ . \\ e_{SL5U,t-365} \\ e_{SL5U,t-366} \end{bmatrix}, \quad \mathbf{e}_{LL1U,t} = \begin{bmatrix} e_{LL1U,t} \\ e_{LL1U,t-1} \\ e_{LL1U,t-2} \\ . \\ . \\ . \\ . \\ . \\ e_{LL1U,t-365} \\ e_{LL1U,t-366} \end{bmatrix}, \quad \mathbf{e}_{LLO,t} = \begin{bmatrix} e_{LLO,t} \\ e_{LLO,t-1} \\ e_{LLO,t-2} \\ . \\ . \\ . \\ . \\ . \\ e_{LLO,t-365} \\ e_{LLO,t-366} \end{bmatrix},$$

$$\mathbf{E}_{SMB,t} = \begin{bmatrix} e_{SMB,t} \\ 0 \\ 0 \\ . \\ . \\ . \\ . \\ . \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{E}_{SMN,t} = \begin{bmatrix} e_{SMN,t} \\ 0 \\ 0 \\ . \\ . \\ . \\ . \\ . \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{E}_{SMO,t} = \begin{bmatrix} e_{SMO,t} \\ 0 \\ 0 \\ . \\ . \\ . \\ . \\ . \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{E}_{SWE,t} = \begin{bmatrix} e_{SWE,t} \\ 0 \\ 0 \\ . \\ . \\ . \\ . \\ . \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\mathbf{E}_{SLO,t} = \begin{bmatrix} e_{SLO,t} \\ 0 \\ 0 \\ . \\ . \\ . \\ . \\ . \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{E}_{SL5U,t} = \begin{bmatrix} e_{SL5U,t} \\ 0 \\ 0 \\ . \\ . \\ . \\ . \\ . \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{E}_{LL1U,t} = \begin{bmatrix} e_{LL1U,t} \\ 0 \\ 0 \\ . \\ . \\ . \\ . \\ . \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{E}_{LLO,t} = \begin{bmatrix} e_{LLO,t} \\ 0 \\ 0 \\ . \\ . \\ . \\ . \\ . \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\Phi_{ii} = \begin{bmatrix} \phi_{ii,1} & \phi_{ii,2} & \phi_{ii,3} & \cdots & \phi_{ii,k_{AR}} & 0 & 0 & \cdots & 0 & 0 & \phi_{ii,364} & \phi_{ii,365} & \cdots & \phi_{ii,367} \\ 1 & 0 & 0 & 0 \cdots & & & & & & & & & 0 & 0 \\ 0 & 1 & 0 & 0 \cdots & & & & & & & & & 0 & 0 \\ 0 & 0 & 1 & 0 \cdots & & & & & & & & & 0 & 0 \\ \cdot & \cdot & & \cdot & & & & & & & & & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & & & & & & & & \cdot & \cdot \\ \cdot & \cdot & & & & \cdot & & & & & & & \cdot & \cdot \\ 0 & 0 & 0 & 0 \cdots & & & & & & & 0 & 0 & 1 & 0 \end{bmatrix}$$

where:

$\phi_{ii,1}, \ldots, \phi_{ii,k_{AR}}, \phi_{ii,364}, \ldots, \phi_{ii,367}$ = the AR parameters that form part of the ARIMA model describing time series i, where some of these may be fixed at zero

k_{AR} = the number of AR parameters at small (non-seasonal) lags used to describe time series i

$$\Phi_{ij} = \begin{bmatrix} \phi_{ij,1} & \phi_{ij,2} & \phi_{ij,3} & \cdots & \phi_{ij,NAR_{ij}} & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & & \cdot & \cdot & \cdot & \cdot & & & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot & \cdot & & & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot & \cdot & & & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

where:

$\phi_{ij,1}, \ldots\ldots\ldots, \phi_{ij,NAR_{ij}}$ = the AR parameters in time series j used in describing time series i, where some of these may be fixed at zero

NAR_{ij} = the maximum lag of the AR parameters from time series j used to describe time series i

$$\Theta_{ii} = \begin{bmatrix} \theta_{ii,1} & \theta_{ii,2} & \theta_{ii,3} & \cdots & \theta_{ii,k_{MA}} & 0 & 0 & \cdots & 0 & 0 & \theta_{ii,363} & \theta_{ii,364} & \cdots & \theta_{ii,367} \\ 1 & 0 & 0 & 0 \cdots & & & \cdots & & & & & & 0 & 0 \\ 0 & 1 & 0 & 0 \cdots & & & \cdots & & & & & & 0 & 0 \\ 0 & 0 & 1 & 0 \cdots & & & \cdots & & & & & & 0 & 0 \\ \cdot & \cdot & & \cdot & & & & & & & & & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & & & & & & & & \cdot & \cdot \\ \cdot & \cdot & & & & \cdot & & & & & & & \cdot & \cdot \\ 0 & 0 & 0 & 0 \cdots & & & \cdots & & & & 0 & 0 & 1 & 0 \end{bmatrix}$$

where:

$\theta_{ii,1}, \ldots, \theta_{ii,k_{MA}}, \theta_{ii,363}, \ldots, \theta_{ii,367}$ = the MA parameters that form part of the ARIMA model describing time series i, where some of these may be fixed at zero

k_{MA} = the number of MA parameters at small (non-seasonal) lags used to describe time series i

$$\Theta_I = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \cdot & \cdot & & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot & \cdot \\ \cdot & \cdot & & & \cdot & \cdot \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

Measurement Equation:

$$SMB_t = [\, 1 \; 0 \; 0 \; \ldots \; 0 \; 0 \,] \, \mathbf{Z}_t$$

3.7.9 Parameter Estimation and Forecasting

A Fortran program ('MULTIEST') was written to estimate the unknown model parameters for VARIMA models in state space form. The following steps were followed in developing the program:

1. Develop software for estimating the parameters of ARIMA models in state space form ('UNIEST').
2. Verify the correct operation of computer program 'UNIEST' by comparing the results obtained using 'UNIEST' with those obtained previously using computer program 'PAREST' (Section 3.6.6).
3. Extend the software to cater for VARIMA models in state space form ('MULTIEST').

It should be noted that the unknown parameters for the multivariate models predicting each of the component time series (i.e. SMB, SMN, SMO, etc.) were estimated separately.

Some of the features of computer programs 'UNIEST' and 'MULTIEST' are given below:

- The initial parameter estimates used were 0.1 in accordance with the recommendations made by Bowermann and O'Connell (1979).
- The model parameters were calculated by maximising the log-likelihood function given in Equation 2.89.
- The IMSL subroutines 'MRRRR', 'LFTRG', 'LFDRG', 'LINRG' and 'MXTYF' were used to perform the matrix operations necessary for the evaluation of the likelihood function (e.g. matrix multiplication and evaluating the determinant and the inverse of a matrix).
- The maximisation of the log-likelihood function was carried out with the aid of the IMSL subroutine 'UMINF'.
- The log-likelihood of the model and the AIC were calculated.
- Reduced forms of the transition matrix (Φ) and the state vector ($\mathbf{Z}_t$) were used in order to increase processing speed. Only the information necessary for evaluating the log-likelihood function was included in the transition matrix and in the state vector (e.g. if the number of unknown parameters for the multivariate model predicting SMB is b, Φ_{SMB} is a [1 x b] matrix and $\mathbf{Z}_{t,SMB}$ is a [b x 1] matrix).

Initially, the following iterative procedure was used to obtain the models with the lowest AIC values for each of the component time series. The model parameters obtained are shown in Tables 3.138 to 3.146.

1. Obtain parameter estimates for the initial models shown in Tables 3.125 and 3.131 to 3.137.
2. Delete any insignificant parameters (i.e. parameters that are small than $2/\sqrt{n}$) and add new parameters where appropriate (i.e. if the parameters at the highest or lowest lags considered are significant, add parameters at higher or lower lags to ensure all significant parameters are included in the model).
3. Obtain parameter estimates for the new models.
4. Delete and add appropriate parameters by examining the values of the new parameters and by comparing the log-likelihood and the AIC of the current models with previous ones.
5. Repeat steps 3 and 4 until there is no further improvement in the AIC.

Next, the 1, 5 and 14 day forecasts of salinity at Murray Bridge were obtained using the models with the lowest AIC values (model C01, Table 3.147). It should be noted that the model used to obtain the multi-step forecasts of salinity at Murray Bridge (model SMB_M_T_20_11_91, Table 3.138) makes use of upstream salinities and flows, which consequently also have to be forecast. Table 3.147 shows the combinations of multivariate models used to obtain multi-step forecasts of salinity at Murray Bridge.

In order to obtain the multi-step forecasts for VARIMA models in state space form, and to convert the differenced forecasts to their non-differenced form, a Fortran program ('MPREDICT') was developed. The following steps were taken in developing the software:

1. Develop software for obtaining multi-step forecasts for ARIMA models in state space form ('PREDICT').
2. Verify the correct operation of computer program 'PREDICT' by comparing the results obtained using 'PREDICT' with those obtained previously using computer program 'FCAST' (Section 3.6.8).
3. Extend the software to cater for VARIMA models in state space form ('MPREDICT').

Table 3.138: Estimates of Model Parameters for Models Predicting Salinity at Murray Bridge (Including MA Parameters)

Vari-able	Lag	Model SMB_M_T_XX_11_91									
		11	12	13	14	15	16	17	18	19	20
SMB	1	0.36	0.37	0.37	0.37	0.38	0.37	0.37	0.37	0.37	0.41
	2										-0.05
	3										-0.07
	364										-0.01
	365	-0.36	-0.36	-0.36	-0.36	-0.35	-0.36	-0.36	-0.36	-0.35	-0.35
	366	0.25	0.24	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.26
	367										-0.01
SMN	1	0.15	0.17	0.17	0.17	0.21	0.17	0.17	0.17	0.17	0.16
	2	0.09	0.08	0.08	0.08		0.08	0.08	0.08	0.08	0.08
	3	0.16	0.17	0.17	0.17	0.19	0.16	0.17	0.17	0.16	0.19
	4	0.02									
SMO	1			0.01							
	2			0.02							
	3			0.00							
	4		0.00								
	5	0.02	0.01								
	6	0.01									
	7	0.02									
SWE	8		0.02								
	9	-0.06	-0.05								
	10	0.13	0.15	0.12	0.12	0.13	0.12	0.12	0.12	0.12	0.12
	11	-0.01									
SLO	9	0.13	0.08	0.09	0.09		0.09	0.09	0.10	0.09	0.09
	10	-0.09	0.03								
	11		-0.13	-0.10	-0.11	-0.10	-0.11	-0.11	-0.11	-0.12	-0.10
	12		0.02								
	13	0.01									
SL5U	11	0.00									
	14		-0.05	-0.03							
	15	0.00	0.05								
	16		-0.04								

Table 3.138 (cont.): Estimates of Model Parameters for Models Predicting Salinity at Murray Bridge (Including MA Parameters)

Vari-able	Lag	Model SMB_M_T_XX_11_91									
		11	12	13	14	15	16	17	18	19	20
LL1U	3					-0.03					
	4					0.10					
	5	0.19			0.23	0.22				0.18	
	6		0.14	0.12	0.08					-0.04	
	7		-0.05								
	8		0.21	0.19	0.18	0.19				0.10	
	9		-0.20	-0.18	-0.21	-0.26				-0.15	
	10		0.03								
	11	-0.09	-0.01								
	12		-0.08	-0.08	-0.13	-0.11				-0.06	
LLO	7					-0.11					
	8					0.34					
	9				-0.18	-0.35			-0.02		
	10	-0.33			-0.13	-0.24			-0.05		
	11		-0.53	-0.54	-0.37	-0.21		-0.41	-0.32		
	12		0.36	0.38	0.37	0.37		0.30	0.26		
	14	0.30									
	15		0.12	0.18	0.13	0.05			0.12		
	16			-0.07	-0.01				-0.06		
e	365	-0.17	-0.17	-0.17	-0.17	-0.15	-0.15	-0.16	-0.16	-0.15	-0.14
	366	0.18	0.16	0.17	0.18	0.17	0.18	0.18	0.18	0.18	0.19
	367	0.09	0.09	0.09	0.09	0.11	0.09	0.10	0.09	0.08	0.11
N^{par}		25	31	24	23	23	12	14	18	17	16
LogL(.)		-2811	-2804	-2805	-2805	-2813	-2811	-2810	-2810	-2809	-2803
AIC		5672	5670	5659	5657	5671	5646	5648	5655	5652	5637

Table 3.139: Estimates of Model Parameters for Models Predicting Salinity at Murray Bridge (No MA Parameters)

Variable	Lag	Model SMB_M_T_XX_11_91						
		21	22	23	24	25	26	27
SMB	1	0.35	0.38	0.38	0.36	0.37	0.38	0.49
	2		-0.08	-0.06				
	3		-0.08	-0.09				
	4		-0.03					
	5		0.01					
	364		0.01	0.01				
	365	-0.32	-0.31	-0.31	-0.32	-0.32	-0.33	-0.34
	366	0.19	0.18	0.18	0.19	0.19	0.19	0.20
	367		-0.01					
SMN	1	0.18	0.16	0.17	0.17	0.20	0.23	
	2	0.07	0.08	0.08	0.07			
	3	0.17	0.17	0.17	0.16	0.18	0.24	
	4		0.08	0.07				
SMO	5		0.02					
	6		0.01					
	7		0.01					
SWE	9		-0.06					0.09
	10	0.13	0.13	0.11	0.11	0.12		0.19
	11		0.01					
SLO	9	0.10						
	10		-0.03					
	11	-0.12						
	13		-0.02					
SL5U	11		0.06					
	15		0.04					
N^{par}		9	23	11	7	6	5	5
LogL(.)		-2826	-2817	-2825	-2837	-2839	-2849	-2885
AIC		5670	5681	5672	5687	5690	5707	5779

Table 3.140: Estimates of Model Parameters for Models Predicting Salinity at Mannum

Variable	Lag	Model SMN_M_T_XX_11_91			
		02	03	04	05
SMN	1	0.45	0.47	0.44	0.46
	2	-0.10	-0.06		
	3	0.05			
	364		0.05		
	365	-0.43	-0.46	-0.44	-0.43
	366	0.18	0.19	0.20	0.21
	367	0.01			
SMO	3	0.04			
	4	0.05			
	5	0.06			
	6	0.05			
	7	0.05			
SWE	6		0.08	0.07	
	7		0.19	0.20	0.30
	8	0.13	0.07	0.06	
	9	-0.02			
	10	0.02			
SLO	11	-0.01			
	12	0.02			
	13	0.09	0.16	0.14	
	14	-0.12	-0.20	-0.18	
	15		0.03		
	16		0.09	0.09	
SL5U	13	0.06			
	14	-0.03			
LL1U	8	0.09			
LLO	12	-0.23			
e	363	-0.01			
	364	0.10	0.17	0.09	
	365		-0.07	-0.07	
	366		0.00		
	367	0.04			
N^{par}		25	15	11	4
LogL(.)		-2851	-2844	-2848	-2864
AIC		5751	5718	5718	5736

Table 3.141: Estimates of Model Parameters for Models Predicting Salinity at Morgan

Variable	Lag	Model SMO_M_T_XX_11_91					
		02	03	04	05	06	07
SMO	1	0.12	0.03				
	2	-0.19	-0.20	-0.21	-0.21	-0.21	-0.21
	3		-0.08	-0.09	-0.09		-0.09
	364		0.07	0.08	0.08		0.08
	365	-0.30	-0.28	-0.28	-0.28	-0.25	-0.29
	366	0.12	0.11	0.11	0.11	0.10	0.11
	367		0.02				
SWE	1		0.39	0.40	0.40	0.38	0.40
	2		0.25	0.26	0.26	0.26	0.26
	3	0.60	0.26	0.27	0.27	0.27	0.27
	4	0.06	0.22	0.22	0.22	0.12	0.22
	5	-0.16	-0.08	-0.07	-0.08		-0.07
	6	-0.01					
SLO	6	0.06					
	7	0.14	0.16	0.13	0.08		0.09
	8	-0.08	-0.08	-0.06			
	9	0.06	0.03				
SL5U	5		0.04				
	6		-0.08	-0.05			
	7	0.30	0.19	0.19	0.17	0.18	0.17
	8	-0.16	-0.12	-0.13	-0.12	-0.10	-0.13
	9	-0.09	-0.05				
LLO	6	0.10	0.12				
	7	-0.02					
	8	0.29	0.27				
	9	-0.59	-0.35	-0.08			
	10		-0.70	-0.71	-0.74	-0.79	
	11		0.72	0.67	0.65	0.68	
e	364	-0.07					
	365	0.47	0.48	0.47	0.47	0.52	0.46
	366	-0.04					
	367	0.07	0.10	0.07	0.07		0.07
N^{par}		23	27	20	17	12	15
LogL(.)		-3364	-3314	-3319	-3319	-3328	-3320
AIC		6774	6682	6677	6672	6681	6670

Table 3.142: Estimates of Model Parameters for Models Predicting Salinity at Waikerie

Variable	Lag	Model SWE_M_T_XX_11_91					
		02	03	04	05	06	07
SWE	1	0.73	0.81	0.81	0.82	0.82	0.86
	2	-0.12	-0.21	-0.20	-0.20	-0.20	-0.18
	3	-0.06					
	364		0.04				
	365		-0.39	-0.37	-0.37	-0.37	-0.46
	366	-0.24	0.21	0.20	0.20	0.20	0.31
	367	0.08	0.00				
SLO	1	0.13	0.15	0.14	0.13	0.13	
	2	0.00					
	3	0.05					
	4	-0.02					
	5	0.08	0.05				
	6		-0.01				
SL5U	3	0.06					
	4	0.03					
	5	0.12	0.12	0.13	0.13	0.13	
	6	-0.03					
LLO	3	-0.33	-0.32	-0.29			
	4	0.41	0.48	0.47			
	5	-0.67	-0.71	-0.71			
	6	0.51	0.50	0.49			
e	364	0.23	0.22	0.19	0.18		
	365	0.27	0.19	0.17	0.18		
	366		-0.07	-0.07	-0.08		
	367		0.02				
N^{par}		20	18	13	9	6	4
LogL(.)		-2813	-2804	-2808	-2811	*)	-2840
AIC		5666	5645	5643	5639		5689

*) Parameters not re-estimated

Table 3.143: Estimates of Model Parameters for Models Predicting Salinity at Loxton

Variable	Lag	Model SLO_M_T_XX_11_91				
		02	03	04	05	06
SLO	1	0.87	0.87	0.89	0.88	0.88
	2	-0.57	-0.56	-0.55	-0.56	-0.56
	3	0.16	0.13	0.13	0.13	0.13
	4	-0.14	-0.07	-0.07	-0.07	-0.07
	5	0.05				
	364		0.06	0.07	0.06	0.06
	365	-0.41	-0.47	-0.47	-0.47	-0.47
	366	0.31	0.33	0.35	0.33	0.33
	367	-0.17	-0.17	-0.16	-0.17	-0.17
SL5U	2	0.21	0.21	0.21	0.20	0.20
	3	-0.01				
	4	0.09	0.10	0.10	0.10	0.10
	5	0.05				
LLO	1	-0.21	-0.21		-0.23	-0.23
	2	0.20	0.23		0.26	0.26
	3	-0.41	-0.39		-0.39	-0.39
	4	0.56	0.52		0.54	0.54
	5	-0.70	-0.66		-0.86	-0.86
	6				0.30	0.30
	7				-0.14	-0.14
e	364	0.00				
	365	0.08	0.09	0.12	0.09	
	366	-0.15	-0.17	-0.14	-0.17	
	367	-0.08	-0.10	-0.07	-0.10	
N^{par}		21	18	13	20	17
LogL(.)		-2541	-2542	-2555	-2541	*)
AIC		5124	5120	5135	5123	

*) Parameters not re-estimated

Table 3.144: Estimates of Model Parameters for Models Predicting Salinity at Lock 5 Upper

Variable	Lag	Model SL5U_M_T_XX_11_91			
		02	03	04	05
SL5U	1	0.56	0.56	0.56	0.56
	2	-0.11	-0.10	-0.10	-0.10
	3	0.02			
	364		0.00		
	365	-0.51	-0.50	-0.50	-0.50
	366	0.30	0.29	0.29	0.29
	367	-0.09	-0.10	-0.10	-0.10
e	364	0.07	0.07	0.07	
	365	-0.03			
	366	0.07	0.07	0.06	
	367	0.03			
N^{par}		10	8	7	5
LogL(.)		-2734	-2734	-2734	*)
AIC		5488	5485	5483	

*) Parameters not re-estimated

Table 3.145: Estimates of Model Parameters for Models Predicting Level at Loxton

Variable	Lag	Model LLO_U_T_XX_11_91		
		02	03	04
LLO	1	1.09	1.09	1.09
	2	-0.36	-0.36	-0.36
	3	0.07	0.07	0.07
	364	-0.04		
	365	-0.35	-0.39	-0.39
	366	0.44	0.44	0.44
	367	-0.14	-0.14	-0.14
e	364	0.00		
	365	-0.33	-0.34	
	366	-0.10	-0.08	
	367	-0.01		
N^{par}		11	8	6
LogL(.)		-1279	-1280	*)
AIC		2580	2575	

*) Parameters not re-estimated

Table 3.146: Estimates of Model Parameters for Models Predicting Level at Lock 1 Upper

Variable	Lag	Model LL1U_M_T_XX_11_91		
		02	03	04
LL1U	1	0.54	0.53	0.53
	2	-0.15	-0.12	-0.12
	3	0.11	0.10	0.10
	4	0.08	0.07	0.08
	5	0.03		
	364	0.10	0.13	0.14
	365	-0.46	-0.45	-0.45
	366	0.17	0.15	0.14
	367	0.01		
LLO	1			-0.09
	2			0.21
	3		0.42	0.27
	4	0.27	-0.17	-0.13
	5	0.14	0.28	0.26
	6	-0.26	-0.34	-0.33
	7		0.07	0.07
e	364	0.17	0.19	0.20
	365	0.30	0.27	0.27
	366	-0.13	-0.15	-0.16
	367	0.03		
N^{par}		16	15	17
LogL(.)		-1759	-1740	-1737
AIC		3549	3509	3508

A reduced form of the transition matrix (Φ) and the state vector ($\mathbf{Z}_t$) were used in order to increase processing speed. Only the information necessary for producing the desired g-step forecast was included (i.e. if the total number of unknown parameters is N^{par}, Φ is a [1 x (N^{par}+g+1)] matrix and $\mathbf{Z}_t$ is a [(N^{par}+g+1) x 1] matrix). The differenced forecasts obtained using 'MPREDICT' were converted to their non-differenced form using the procedure outlined in Section 3.6.8.

However, as the ultimate objective is to obtain the best forecasts of salinity at Murray Bridge, other combinations of models were tried in attempt to achieve this goal (Table

3.147). The model parameters for the multivariate component models shown in Table 3.147 are shown in Tables 3.138 to 3.146.

Table 3.147: Combinations of Models Used to Obtain Multi-Step Forecasts of Salinity at Murray Bridge

	Combined model											
Component model	C01	C02	C03	C04	C05	C06	C07	C08	C09	C10	C11	C12
SMB_M_T_XX_11_91	20	20	12	16	21	21	23	24	25	26	27	25
SMN_M_T_XX_11_91	04	03	04	04	04	05	05	05	05	05		05
SMO_M_T_XX_11_91			07									
SWE_M_T_XX_11_91	05	03	05	05	05	06	06	06	06	06	06	07
SLO_M_T_XX_11_91	03	02	03	03	03	06	06	06	06	06	06	
SL5U_M_T_XX_11_91	04	02	04	04	04	05	05	05	05	05	05	
LL1U_M_T_XX_11_91			04									
LLO_M_T_XX_11_91	03	02	03	03	03	04	04	04	04	04	04	

In model C02, the component model for forecasting salinity at Murray Bridge with the lowest AIC value was retained. However, the component models with the lowest AIC for forecasting the upstream time series were replaced with those with the highest log likelihood.

In model C03, the component model with the highest log likelihood was used for forecasting salinity at Murray Bridge, while the component models with the lowest AIC values were used to forecast the upstream time series.

The component models used in model C04 were the same as those used in model C03, with the exception that the component model for forecasting salinity at Murray Bridge had fewer parameters.

In models C05 to C12, the MA terms were omitted from the component models forecasting salinity at Murray Bridge. The component models used in model C05 were identical to those used in model C04, with the exception that the MA parameters for the model forecasting salinity at Murray Bridge were omitted.

The component model for forecasting salinity at Murray Bridge was the same in models C05 and C06. However, changes were made to the component models forecasting the upstream time series. The major change was the omission of the MA parameters. It should be noted that in most instances, the parameters for these models were not re-

estimated, and consequently, no values for the log likelihood and AIC are shown in Tables 3.142 and 3.146.

The only difference between models C06 to C11 is that different AR parameters were used in the component model forecasting salinity at Murray Bridge. In model C10, the effect of omitting inputs of salinity at Waikerie was assessed, and in model C11, the effect of omitting inputs of salinity at Mannum was assessed.

In model C12, the parameters for the component model forecasting salinity at Waikerie, which does not have any MA parameters, were re-estimated.

3.7.10 Results / Discussion

The 1, 5 and 14 day forecasts were examined to assess each model's short-term and longer-term forecasting ability. The RMSE, AAPE and AAE between the actual and predicted salinities for the 1, 5 and 14 day forecasts are shown in Tables 3.148 and 3.149.

Table 3.148: Results for Combined Models C01 to C04 (Models With MA Parameters for Predicting Salinity at Murray Bridge)

Model	Forecast period (days)	RMSE (EC units)	AAPE (%)	AAE (EC units)
C01	1	9.9	1.12	6.8
	5	43.5	5.29	31.0
	14	126.1	15.3	87.8
C02	1	9.9	1.12	6.8
	5	43.5	5.29	31.0
	14	127.3	15.39	89.7
C03	1	10.1	1.17	6.9
	5	45.3	5.44	31.5
	14	125.8	15.26	86.2
C04	1	10.0	1.14	6.8
	5	44.5	5.33	31.0
	14	124.9	15.03	85.4

Table 3.149: Results for Combined Models C05 to C12 (Models Without MA Parameters for Predicting Salinity at Murray Bridge)

Model	Forecast period (days)	RMSE (EC units)	AAPE (%)	AAE (EC units)
C05	1	9.7	1.12	6.7
	5	42.8	5.13	29.2
	14	121.2	14.6	80.9
C06	1	9.7	1.12	6.7
	5	38.9	4.87	27.5
	14	107.4	13.8	74.7
C07	1	9.6	1.08	6.6
	5	43.8	4.99	28.7
	14	110.0	13.8	74.2
C08	1	9.5	1.05	6.3
	5	37.9	4.52	25.8
	14	104.4	13.0	71.0
C09	1	9.6	1.05	6.4
	5	37.8	4.49	25.7
	14	104.4	12.9	70.7
C10	1	9.5	1.07	6.5
	5	37.5	4.71	26.3
	14	105.9	13.5	72.6
C11	1	10.9	1.15	7.1
	5	44.2	5.19	30.8
	14	121.0	14.12	82.4
C12	1	9.6	1.05	6.4
	5	37.7	4.46	25.6
	14	102.9	12.7	69.8

The following general comments can be made by examining the prediction errors and the plots of forecasts for the various tests.

1. As expected, the forecast accuracy decreases as the forecast period increases. As shown in Tables 3.148 and 3.149, all one day forecasts are very good. However, as the one day forecast is used to produce the two day forecast and so on, any small errors in

the one day forecast are amplified as the forecast length increases. This is demonstrated in Figures 3.147 to 3.149, and is typical for all models.

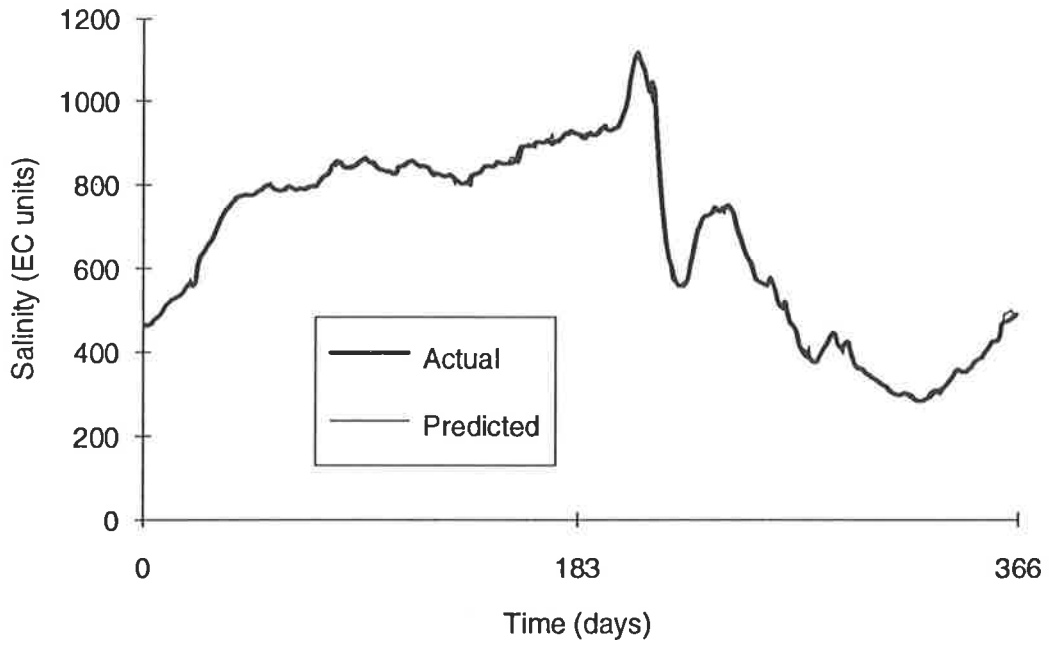


Figure 3.147: Actual and Predicted Salinities - Combined Model C12 - 1 Day in Advance

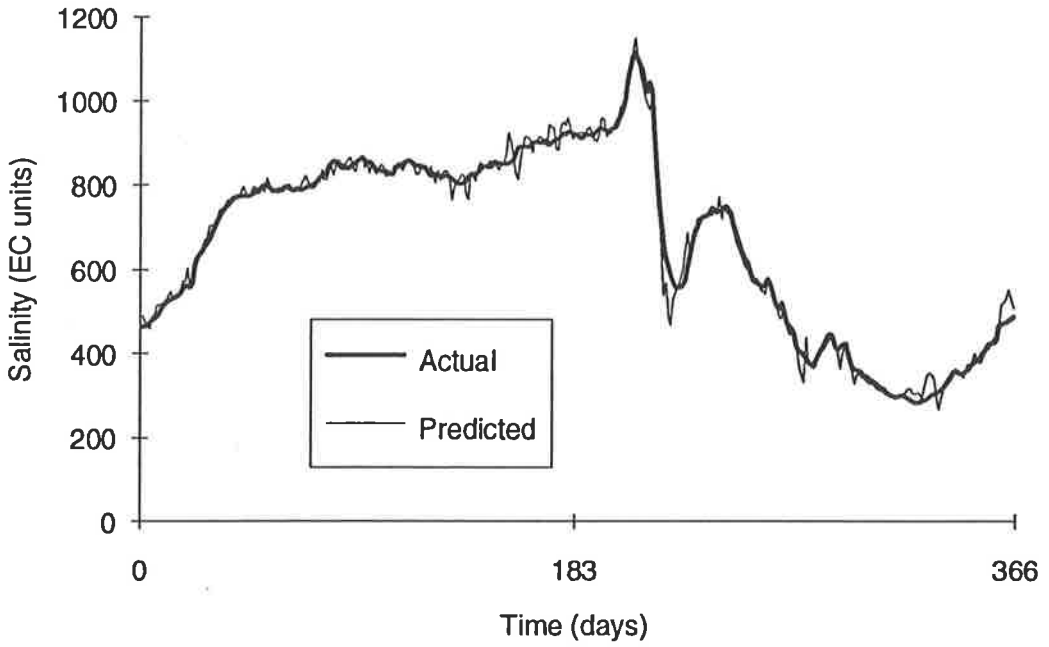


Figure 3.148: Actual and Predicted Salinities - Combined Model C12 - 3 Days in Advance

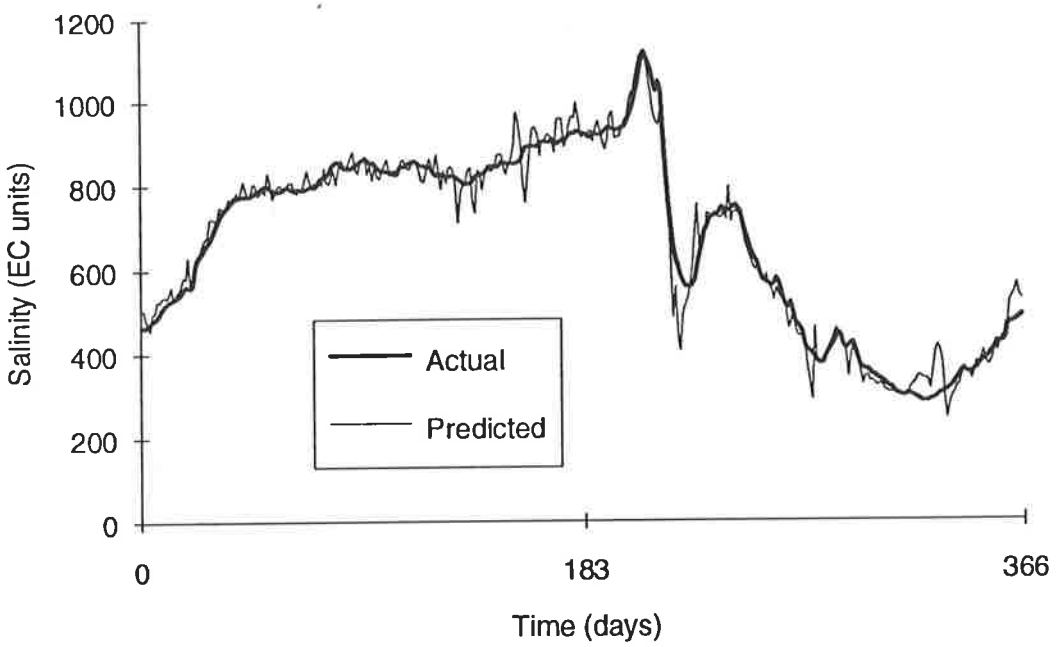


Figure 3.149: Actual and Predicted Salinities - Combined Model C12 - 5 Days in Advance

2. Too many parameters induce noise, resulting in an increase in prediction error as illustrated in Table 3.150.

Table 3.150: Change in Prediction Error with the Number of Parameters

Model	Number of parameters			RMSE (EC units)
	AR	MA	Total	(14 day forecast)
C02	69	22	91	127.3
C01	54	15	69	126.1
C04	50	15	65	124.9
C05	50	12	62	121.2
C07	49	0	49	110.0
C06	47	0	47	107.4
C09	44	0	44	104.4
C12	14	0	14	102.9

However, if one of the dominant parameters is omitted, the prediction error is increased as indicated in Table 3.151. In model C10, salinity at Waikerie was omitted whereas salinity at Mannum was omitted in model C11.

Table 3.151: Change in Prediction Error with the Omission of Dominant Parameters

Model	Number of parameters			RMSE (EC units) (14 day forecast)
	AR	MA	Total	
C08	45	0	45	104.4
C10	43	0	43	105.9
C11	39	0	39	121.0

3. For the data set used, the AIC was not a good indicator of model performance, as illustrated in Table 3.152. Better values of AIC were obtained for models having a greater number of parameters, indicating that a better fit was obtained to the data used to estimate the model parameters when more parameters were used. However, when using too many parameters, overfitting occurred, resulting in poor generalisation ability, as indicated by the greater RMS forecasting errors (Table 3.152).

Table 3.152: Performance of Models with Different AIC Values

Time series predicted	AIC		
	Model		
	C01	C04	C05
SMB	5637	5646	5670
SMN	5718	5718	5718
SWE	5639	5639	5639
SLO	5120	5120	5120
SL5U	5485	5485	5485
LLO	2575	2575	2575
N^{par}	69	65	62
RMSE (EC units) (14 day forecast)	126.1	124.9	121.2

4. A plot of the best 14-day forecast is shown in Figure 3.150. It is very similar to the best 14-day forecast obtained using the univariate time series (ARIMA) model (Figure 3.144). Both forecasts are very noisy, although there are no appreciable lags in the forecast compared to the actual data. However, the excessive noise renders the forecast useless, as it is impossible to discern which of the large variations in predicted salinity will actually occur. As discussed above, the noise present is a result of the recursive procedure used to obtain longer term forecasts. The marked drop in salinity in the

vicinity of day 125 can be attributed to a sharp drop in the salinity at the same time in the previous year (see Figure 3.144).

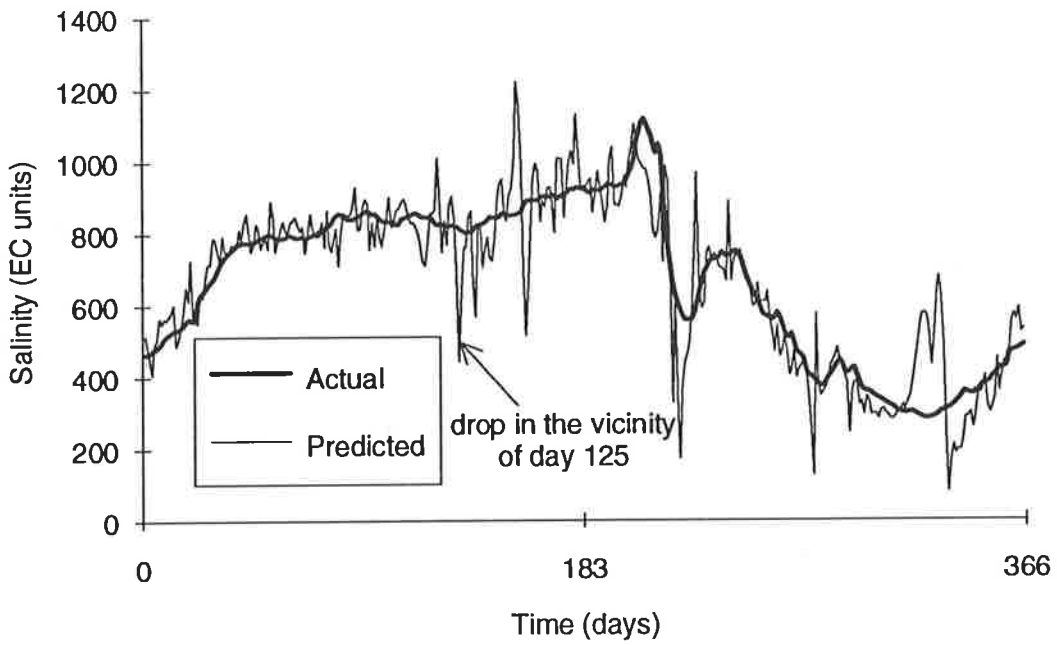


Figure 3.150: Actual and Predicted Salinities - Combined Model C12 - 14 Days in Advance

The best forecast was obtained using model C12, which has by far the smallest number of parameters. In this model, salinity at Murray Bridge (SMB) was predicted using only its own past values as well as upstream values of salinity at Mannum (SMN) and Waikerie (SWE). Salinity at Mannum was predicted with the aid of its own past values as well as past values of salinity at Waikerie, and salinity at Waikerie was predicted using its own past values only.

5. As discussed above, the 14 day forecasts produced are very noisy. It is possible to remove some of that noise by filtering. Filtering was carried out by applying the one-sided linear filter described in Section 3.6.9 to the 14 day forecast obtained using model C12.

Table 1.153 indicates that filtering results in a considerable improvement in the forecasts obtained. A comparison of the plots of the filtered and unfiltered forecasts (Figures 3.150 and 3.151) shows that the filtered 14 day forecast is considerably smoother. However, the forecast is still noisy, and hence it is still difficult to discern which of the predicted variations in salinity will actually occur.

Table 3.153: Filtered 14 Day Forecasts for Combined Model C12 (One-sided Filter)

ξ	RMSE (EC units)	AAPE (%)	AAE (EC units)
0	102.9	12.7	69.8
1	95.4	12.0	64.5
2	91.1	11.5	61.2
4	85.5	10.9	57.5

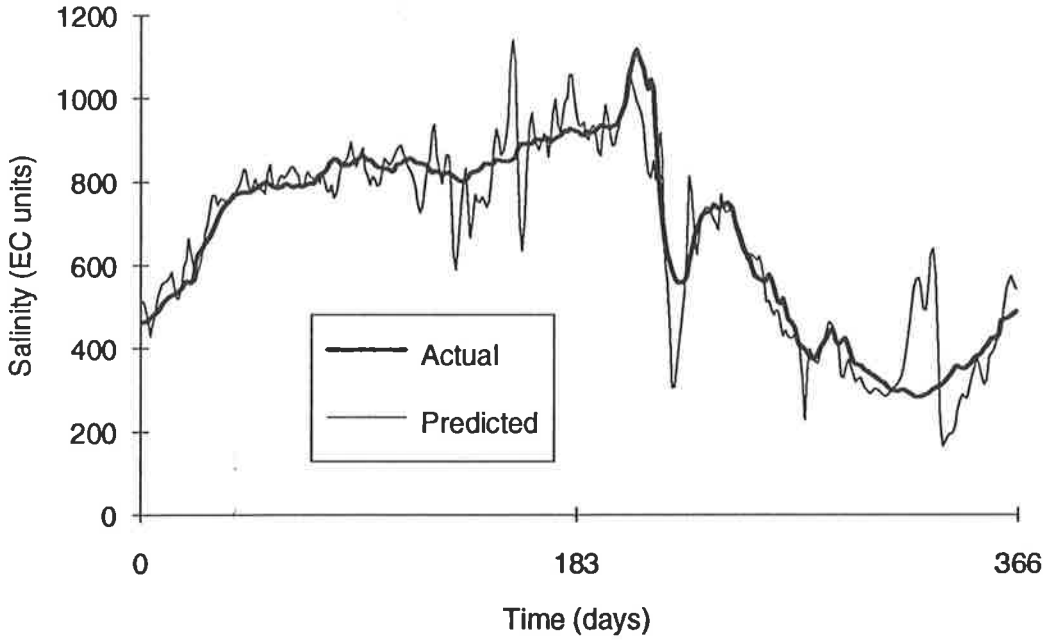


Figure 3.151: Actual and Predicted Salinities (Filtered) - Combined Model C12 - 14 Days in Advance ($\xi = 4$)

It has been shown that filtering can be used to improve the 14 day forecasts obtained, but not to the extent that the forecast becomes useful. It should be noted, however, that only one type of filter was used here to demonstrate the positive effect that filtering has on the results. The results could be further improved by applying other, possibly different types of, filters.

3.7.11 Diagnostic Checking

The residuals of the models used for forecasting were obtained using computer program 'MPREDICT'. A plot of the residuals (not shown here) showed that there were no outliers and there was no apparent trend. In addition, the mean of the residuals was close to zero. The variance of the residuals was found to be constant, and the histogram

of the residual frequencies (not shown here) was normally distributed, indicating that there is no need to transform the original data using a Box-Cox transform (Hipel, 1985).

The Box-Pierce method (Box and Pierce, 1970) was then used to examine the residuals. The ACF and the PACF of the residuals was calculated using computer program 'APCORR' and checked for any significant spikes. There were no significant spikes at low lags (i.e. 1 to 10) and no significant spikes at seasonal lags (i.e. 360 to 370), suggesting that the residuals are uncorrelated. There were occasional significant values, all of which were close to the critical value for significance. As discussed in Section 3.6.6, such values do not suggest strong correlations and may be ignored.

The Portmanteau lack-of-fit test (Bowermann and O'Connell, 1979) was also used to check the whiteness of the residuals. The Box-Pierce chi-square statistic, Q^{BP}, is the most well known test statistic for the Portmanteau lack-of-fit test and was used here. Q^{BP} was evaluated as part of computer program 'APCORR'. A value of k' = 3L was used for reasons discussed in Section 3.6.7. The values of Q^{BP} and $\chi^2_{(5)}$ for the various models are shown in Table 3.154.

As can be seen from Table 3.154, all models are adequate. However, the cross-correlations between the residuals of the various component models used in the combined models also have to be examined (Harvey, 1981). This was done for model C12 only, as it gave the best forecasts. The values of Q^{BP} and $\chi^2_{(5)}$ for the various cross-correlations between the residuals are shown in Table 3.155. The cross-correlations between the residuals were obtained with the aid of computer program 'CROSSCORR'. A value of k' = 365 was used to evaluate Q^{BP}. It can be seen that all values of Q^{BP} are less than the corresponding value of $\chi^2_{(5)}$, indicating that model C12 is adequate.

Harvey (1981) suggests that a Q^{BP} matrix should be assembled to assess the adequacy of a multivariate time series model. Such a matrix contains the Q^{BP}-values for the autocorrelations of the residuals in the diagonal elements and the Q^{BP}-values of the cross-correlations between the residuals in the off-diagonal elements. The Q^{BP} matrix for model C12 is given in Table 3.156. The numbers in the top line of each cell represent the value of $Q^{BP}(365)$, while the value in brackets underneath represents the allowable chi-squared value. As can be seen, all values of Q^{BP} are less than their allowable value, indicating that all the residuals are uncorrelated and that model C12 is acceptable.

Table 3.154: Values of Q^{BP} and $\chi^2_{(5)}$ for the Multivariate Models Used for Forecasting

Model	m'	$Q^{BP}(365)$	$\chi^2_{(5)}$
SMB_M_T_12_11_91	342	300	386
SMB_M_T_16_11_91	353	301	397
SMB_M_T_20_11_91	349	282	393
SMB_M_T_21_11_91	356	293	401
SMB_M_T_25_11_91	359	290	404
SMN_M_T_03_11_91	350	277	394
SMN_M_T_04_11_91	354	340	399
SMN_M_T_05_11_91	361	266	406
SMO_M_T_03_11_91	338	313	382
SMO_M_T_07_11_91	350	308	394
SWE_M_T_03_11_91	347	295	391
SWE_M_T_05_11_91	356	290	401
SWE_M_T_06_11_91	359	290	404
SWE_M_T_07_11_91	361	307	406
SLO_M_T_02_11_91	344	259	388
SLO_M_T_03_11_91	347	274	391
SLO_M_T_06_11_91	348	274	392
SL5U_M_T_02_11_91	355	249	400
SL5U_M_T_04_11_91	358	249	403
SL5U_M_T_05_11_91	360	245	405
LL1U_M_T_04_11_91	348	335	392
LLO_U_T_02_11_91	354	284	399
LLO_U_T_03_11_91	357	285	402
LLO_U_T_04_11_91	359	319	404

Table 3.155: Values of Q^{BP} and $\chi^2_{(5)}$ for the Cross-Correlations Between the Residuals For Combined Model C12

Models	m'	$Q^{BP}(365)$	$\chi^2_{(5)}$
SMB_M_T_25_11_91/SMN_M_T_05_11_91	355	276	400
SMB_M_T_25_11_91/SWE_M_T_07_11_91	355	240	400
SMN_M_T_05_11_91/SMB_M_T_25_11_91	355	274	400
SMN_M_T_05_11_91/SWE_M_T_07_11_91	357	270	402
SWE_M_T_07_11_91/SMB_M_T_25_11_91	355	309	400
SWE_M_T_07_11_91/SMN_M_T_05_11_91	357	353	402

Table 3.156: Q^{BP}-matrix for Combined Model C12

	1.	2.	3.
1. SMB_M_T_25_11_91	290 (404)	276 (400)	240 (400)
2. SMN_M_T_05_11_91	274 (400)	266 (406)	270 (400)
3. SWE_M_T_07_11_91	309 (400)	353 (402)	307 (406)

3.7.12 Summary / Conclusions

Differencing the multivariate time series using a degree of seasonal differencing and a degree of non-seasonal differencing of 1 (i.e. D = 1 and d = 1) resulted in joint stationarity of the time series. The method of Haugh and Box (1977) was used to identify the initial order of the multivariate model.

The models developed were found to be sensitive to the number of model parameters. The presence of less-significant model parameters increased model noise, resulting in decreased performance. The best forecasts were obtained by using combined model C12, which is shown in Equations 3.11 to 3.13.

$$u_{SMB,t} = (0.37)\, u_{SMB,t-1} - (0.32)\, u_{SMB,t-365} + (0.19)\, u_{SMB,t-366} + (0.20)\, u_{SMN,t-1} + (0.18)\, u_{SMN,t-3} + (0.12)\, u_{SWE,t-10} + e_{1t} \quad (3.11)$$

$$u_{SMN,t} = (0.46)\, u_{SMN,t-1} - (0.43)\, u_{SMN,t-365} + (0.21)\, u_{SMN,t-366} + (0.30)\, u_{SWE,t-7} + e_{2t} \quad (3.12)$$

$$u_{SWE,t} = (0.86)\, u_{SWE,t-1} - (0.18)\, u_{SWE,t-2} - (0.46)\, u_{SWE,t-365} + (0.31)\, u_{SWE,t-366} + e_{3t} \quad (3.13)$$

The Box-Pierce method and the Portmanteau lack-of-fit test indicated the adequacy of the above model.

The one day forecasts produced by model C12 were very good, with a RMSE of 9.6 EC units and an AAPE of 1.1% for 1991. However, the 14 day forecasts were very noisy, with a RMSE of 102.9 EC units and an AAPE of 9.6% for 1991. The 14 day forecasts

were able to be improved significantly (RMSE = 85.5 EC units, AAPE = 10.9%) by using a one-sided filter.

3.8 Comparison of Results

3.8.1 Introduction

In this section, the performances of the univariate ANN (UANN), multivariate ANN (MANN), univariate time series (ARIMA) and multivariate time series (VARIMA) models are compared in terms of generalisation ability and ease of use. For reasons explained in Section 3.3.5, the RMSE was used to compare the generalisation ability of the various models for forecasting periods of 1, 5 and 14 days. The performance of the models was also compared with that of a naive model (i.e. $z_{t+g} = z_t$, where g = the forecasting period). In order to obtain a fair comparison between the VARIMA and MANN models, MANN models using forecast flows and river levels as inputs were not considered in this section.

3.8.2 Artificial Neural Network Models

The RMSEs of the 1, 5 and 14 day forecasts obtained using the best UANN model (model SMB_U_A_15_06_91), the best MANN model (model SMB_M_A_79_13_91) and the naive model are shown in Figure 3.152. It should be noted that both ANN models use the raw data and the direct method of forecasting. The results obtained indicate that:

- The MANN model performs better than the UANN model for all forecasting periods, suggesting that there is a definite advantage in using causal variables when forecasting salinity in the River Murray at Murray Bridge. This is especially true for longer forecasting periods, as indicated by the increasing gap between the forecasts obtained using the UANN and MANN models with increasing forecasting period.
- The 1-day forecasts obtained using the UANN and MANN models are worse than that obtained using the naive model, suggesting that ANN models are not suited to short-term forecasting.
- As the forecasting period increases, the ANN models perform significantly better than the naive model, indicating that ANN models are suited to longer term forecasting. A likely reason for this is that ANN models base their forecasts on the approximate underlying relationships that generated the data.

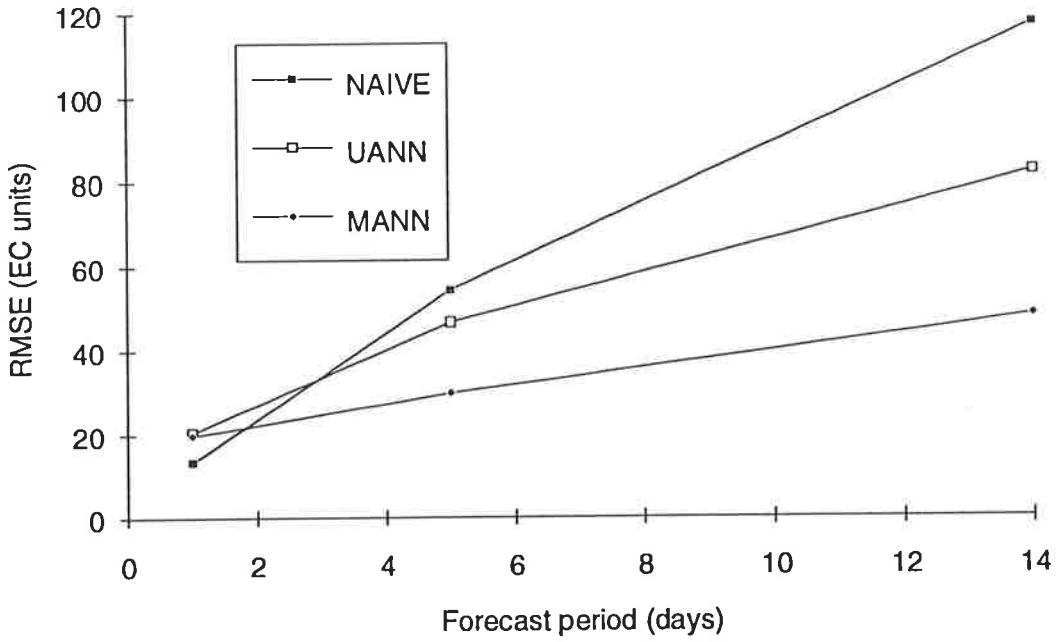


Figure 3.152: Comparison of the RMS Forecasting Errors Obtained for 1991 Using the Best UANN and MANN models (Direct Method of Forecasting) as well as a Naive Model

3.8.3 Time Series Models

The RMSEs of the 1, 5 and 14 day forecasts obtained using the best ARIMA model (model SMB_U_T_04_11_91), the best VARIMA model (model C12) and the naive model are shown in Figure 3.153. It can be seen that:

- The ARIMA and VARIMA models perform significantly better than the naive model for all forecasting periods.
- The VARIMA model performs only marginally better than the ARIMA model, suggesting that the benefit of using causal variables is only minimal. The reason for this is that in ARMA type models, the model coefficients are estimated by maximising the likelihood function for the one day forecast, for which causal variables are less critical.

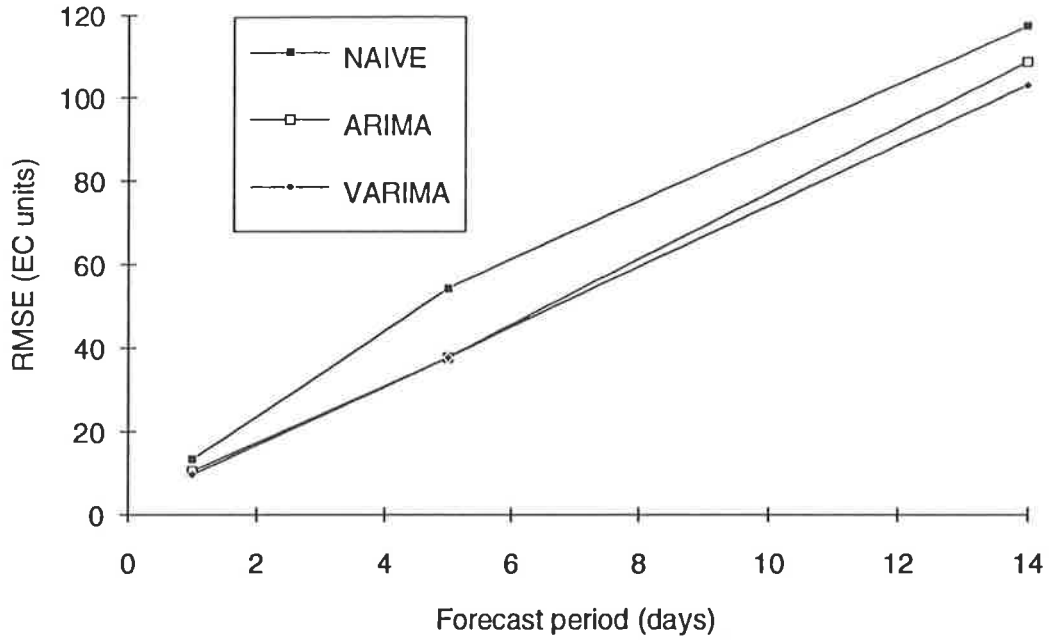


Figure 3.153: Comparison of the RMS Forecasting Errors Obtained for 1991 Using the Best ARIMA, VARIMA and Naive Models

It should be noted that the performance of the ARIMA and VARIMA models can be improved considerably by filtering the forecasts obtained, as shown in Figure 3.154 for the 14 day forecast.

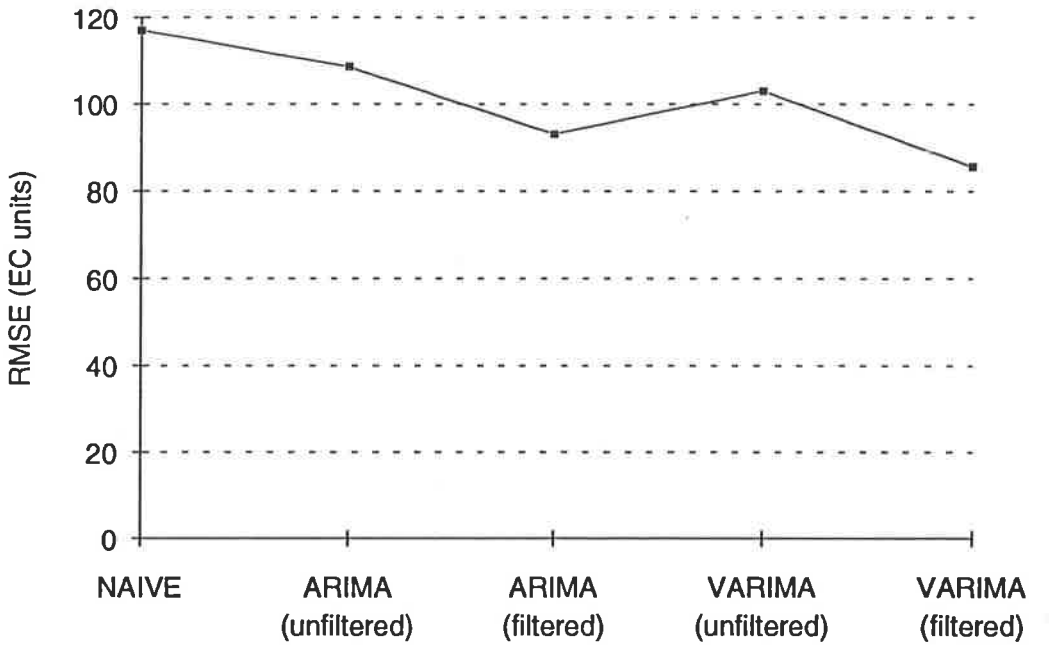


Figure 3.154: Effect of Filtering on the 14-Day Forecasts (1991) Obtained Using the ARIMA and VARIMA Models

3.8.4 Univariate Models

The RMSEs of the 1, 5 and 14 day forecasts obtained using the best ARIMA model (model SMB_U_T_04_11_91), the best UANN model (model SMB_U_A_15_06_91) and the naive model are shown in Figure 3.155. It can be seen that:

- The performance of the ARIMA model is significantly better than that of the naive model for all forecasting periods.
- The UANN model is not suited to short term forecasting, as it performs worse than the naive model.
- The UANN model is suited to longer term forecasting, as it performs better than the ARIMA and naive models.

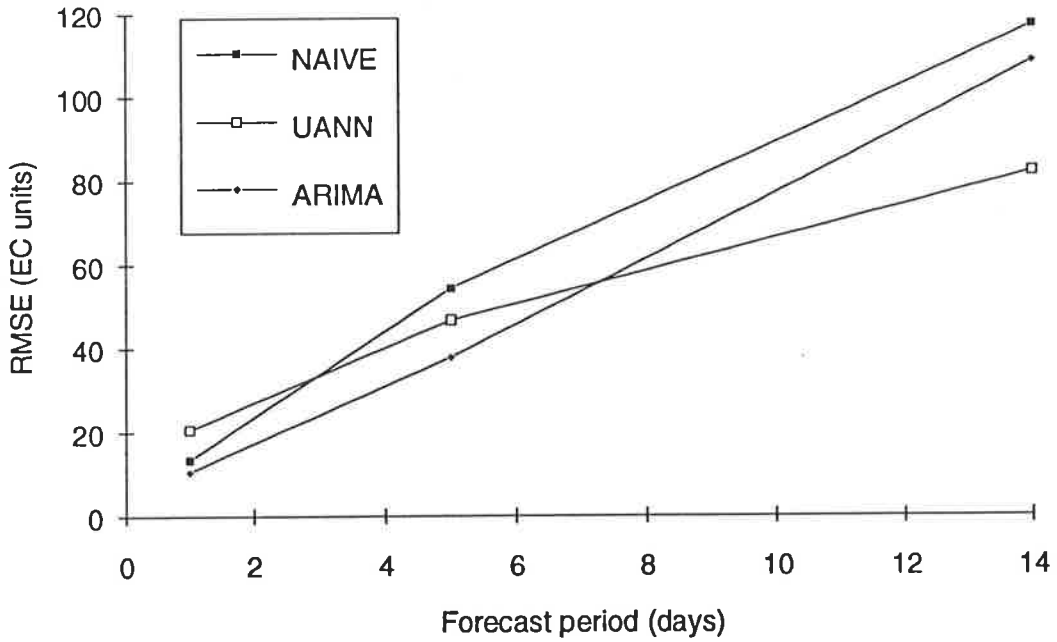


Figure 3.155: Comparison of the RMS Forecasting Errors Obtained for 1991 Using the Best UANN (Direct Method of Forecasting), ARIMA and Naive Models

It should be noted that after filtering, the performance of the ARIMA model approaches that of the UANN model for a forecasting period of 14 days, as indicated in Figure 3.156.

The UANN model used in the above comparison utilised the raw data in conjunction with the direct method of forecasting. The behaviour of such a network is characterised by the effect the hidden layer nodes have on the raw data. However, when the differenced data are used in conjunction with the recursive forecasting method, the

behaviour of UANN models is very similar to that of ARIMA models. This can be demonstrated by comparing models SMB_U_A_04_02_91 and SMB_U_T_04_11_91. Model SMB_U_A_04_02_91 operates in a manner similar to ARIMA models, as it does not have any hidden layers and uses a linear transfer function. In both models, the outputs are a linear combination of the inputs. Consequently, one would expect the coefficients to be very similar for both models. A comparison between the coefficients obtained using model SMB_U_T_04_11_91 and the connection weights obtained for model SMB_U_A_04_02_91 is shown in Table 3.157. It can be seen that the magnitude of the coefficients for the ARIMA model is between 1.25 and 1.35 times that of the connection weights for the UANN model. However, the ratio is very consistent, indicating that the relationship between the input variables determined by both models is very similar.

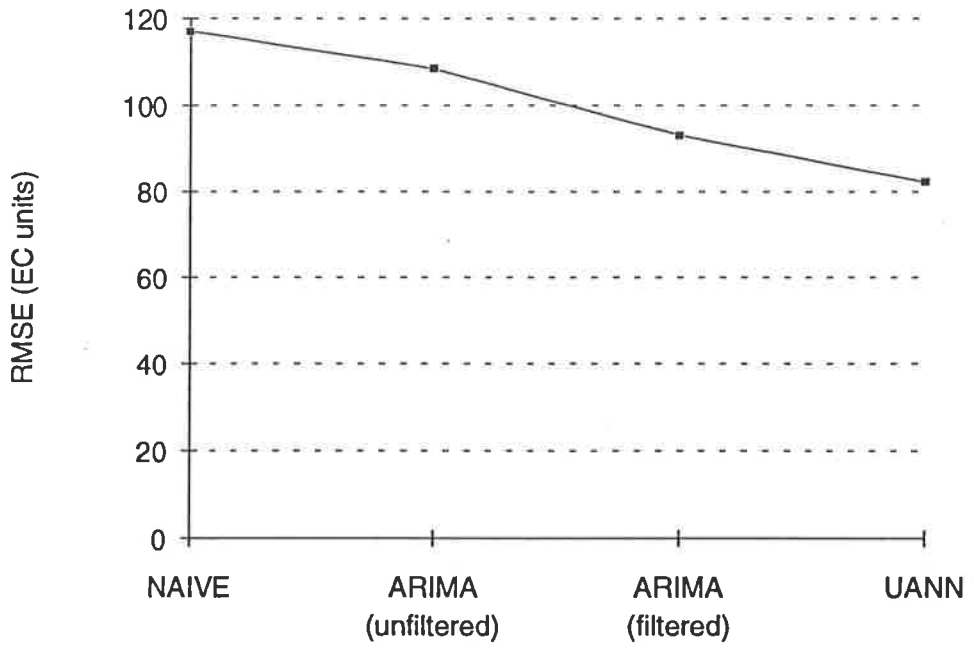


Figure 3.156: Effect of Filtering on the 14-Day Forecasts (1991) Obtained Using the ARIMA Model

Table 3.157: Comparison of the Coefficients Obtained for Model SMB_U_T_04_11_91 with the Weights Obtained for Model SMB_U_A_04_02_91

Lags of inputs	Coefficients for ARIMA model SMB_U_T_04_11_91	Weights for ANN model SMB_U_A_04_02_91	Ratio (ARIMA/UANN)
1	+0.6240	+0.4979	1.25
365	-0.3790	-0.2815	1.35
366	+0.2253	+0.1780	1.27

The 1 and 14 day forecasts obtained using models SMB_U_T_04_11_91 and SMB_U_A_04_02_91 (Table 3.158) indicate that the ARIMA model clearly outperforms the UANN model when they operate in a similar manner.

Table 3.158: Comparison of 1 and 14 Day Forecasts Obtained Using Models SMB_U_T_04_11_91 and SMB_U_A_04_02_91

Forecasting period (days)	RMSE (EC units)	
	Model SMB_U_T_04_11_91	Model SMB_U_A_04_02_91
1	10.5	14.9
14	108.5	123.8

3.8.5 Multivariate Models

The RMSEs of the 1, 5 and 14 day forecasts obtained using the best MANN model (model SMB_M_A_79_13_91), the best VARIMA model (model C12) and the naive model are shown in Figure 3.157. It can be seen that:

- The performance of the VARIMA model is considerably better than that of the naive model for all forecasting periods.
- The performance of the MANN model is worse than that of the VARIMA and the naive models for a forecasting period of 1 day, indicating that MANN models are not suited for short term forecasting.
- As the forecasting length increases, the MANN model performs vastly better than the VARIMA and naive models, suggesting that MANN models are suited to longer term forecasting.

Even after filtering, the 14 day forecast obtained using the VARIMA model is still considerably worse than that obtained using the MANN model, as shown in Figure 3.158.

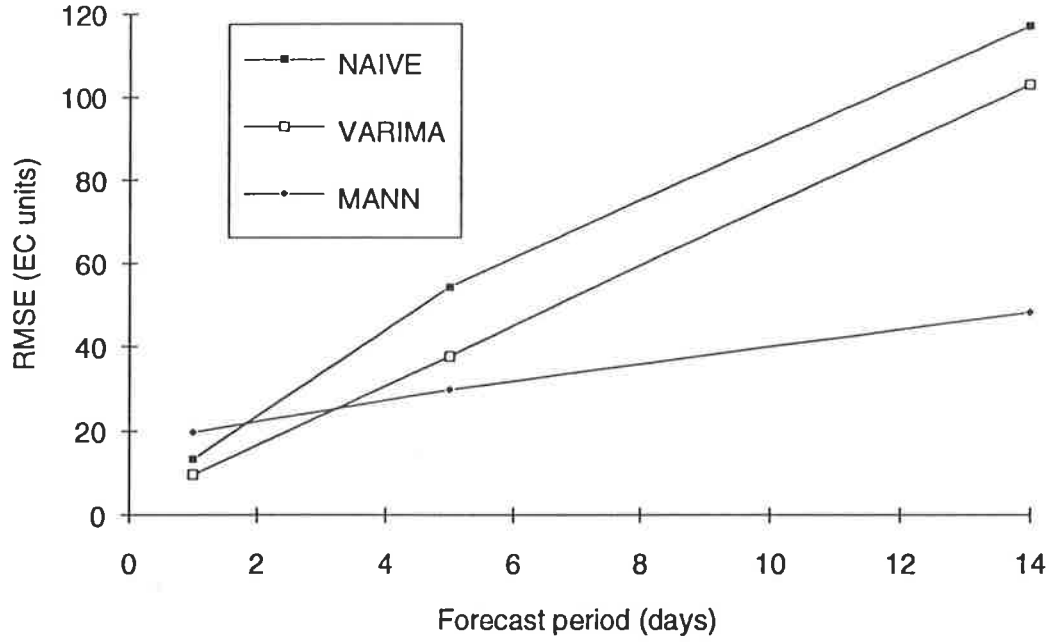


Figure 3.157: Comparison of the RMS Forecasting Errors Obtained for 1991 Using the Best MANN, VARIMA and Naive Models

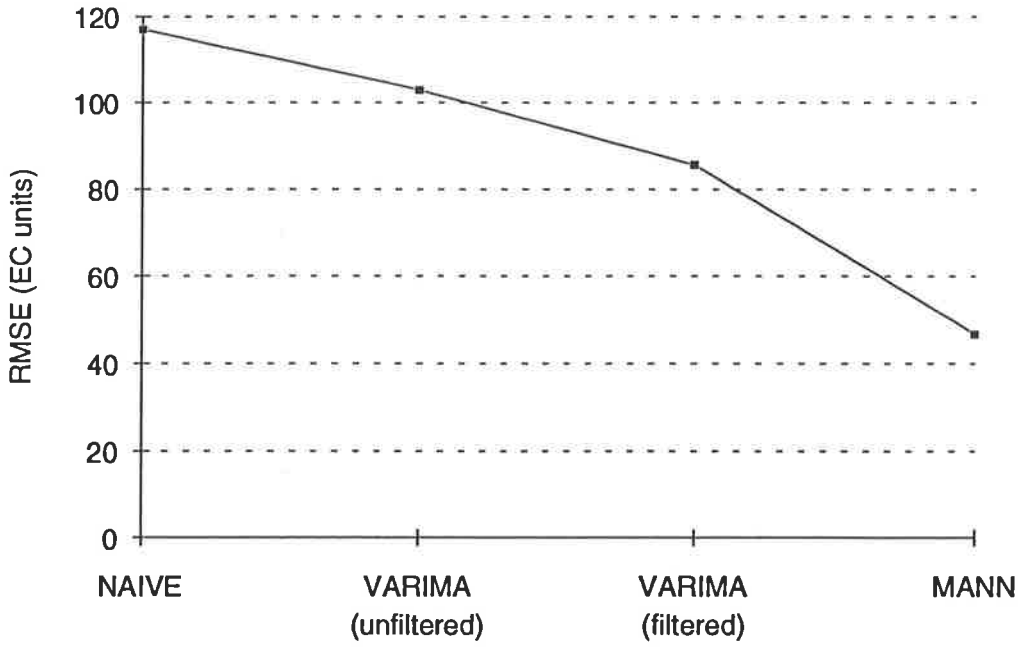


Figure 3.158: Effect of Filtering on the 14-Day Forecasts (1991) Obtained Using the VARIMA Model

3.8.6 Artificial Neural Network Versus ARMA Type Models

The results obtained indicate that ARMA type models are better suited to short-term forecasting and that ANN models are better suited to longer-term forecasting. Possible reasons for this include that ANN models base their forecasts on the underlying relationships that generated the time series, and that they can be trained for a specific forecasting period directly, without having to rely on a recursive forecasting procedure. However, it should be noted that the performance of different types of models also depends on the type and complexity of the data used.

Of the two types of models considered, the ANN models were far simpler to develop. When developing the ANN models, there was no need to difference the data, as ANNs are able to account for non-stationarities in the data (e.g. trends, seasonal variation) with the aid of their hidden layer nodes. In addition, using the direct method for obtaining multi-step forecasts is simpler than using the recursive method.

The research carried out indicates that the forecasts obtained using ANN models are relatively insensitive to the number of model inputs. In contrast, the performance of ARMA type models is substantially reduced by the presence of less-significant inputs, which induce model noise.

The results obtained indicate that the generalisation ability of ANN models is superior to that of ARMA type models when limited data sets are available. The VARIMA models that were found to be superior using Akaike's Information Criterion did not produce the best forecasts when an independent data set was used. In other words, the models that provided the best fit to the training data, even when parsimony considerations were taken into account, did not generalise as well as models that used fewer model parameters, although the latter provided an inferior fit to the training data. One would expect this situation to be rectified if more training data were available, which would enable the model to determine a relationship that is more general. As mentioned above, such overfitting problems did not occur with the ANN models.

A potential advantage of using ARMA type models is that a direct mathematical relationship between the input and output variables is obtained. However, by carrying out a sensitivity analysis, the magnitude and the strength of the relationship between the model inputs and the model outputs that has been determined by an ANN model can be easily obtained.

An advantage of ARMA type models is that the procedure for their development, in particular the input identification stage, is much better defined. Although the analytical method proposed by Haugh and Box (1977) is very elaborate and time consuming, it provides a methodology for finding appropriate inputs to multivariate ARMA type models, which can be used even when no *a priori* knowledge is available to suggest possible inputs. ANN models rely on empirical methods to determine appropriate model inputs, which generally results in an excessive number of inputs. Although ANNs are relatively insensitive to the number of model inputs, excessive numbers of inputs significantly increase network size, making it difficult to optimise the connection weights and increasing processing time.

Another difficulty associated with the development of ANN models is choosing an appropriate network geometry and optimising the connection weights, as discussed in Section 3.5.3.

3.9 Longer Term Salinity Forecasts

3.9.1 Introduction

In this section, a MANN model is developed to forecast salinity in the River Murray at Murray Bridge 28 days in advance, as MANN models are the model type most suitable for longer-term forecasts (see Section 3.8). As discussed in Section 3.3.2, forecasts of this length are desirable when making adjustments to the schedule for pumping water from Murray Bridge to Adelaide, if salinity considerations are to be taken into account.

3.9.2 Determination of Model Inputs

The model inputs were based on those used in Section 3.5.2 for forecasting salinity at Murray Bridge 14 days in advance (training / testing set 10 - Table 3.38). It was decided to omit the inputs of level at Lock 1 Lower, as there is a strong correlation between the flow and level data (Section 3.7.6.2). It was also decided to include inputs of salinity at Lock 5 Upper, as Lock 5 Upper is the site furthest upstream for which data were available, and using data from sites further upstream is desirable for longer term forecasts, as a result of the longer travel times. However, as can be seen by comparing Figures 3.12, 3.17, 3.22, 3.27, 3.32 with Figures 3.16, 3.21, 3.26, 3.31, 3.36 respectively, at sites further upstream, the mean salinity is considerably lower, and only the major variations in salinity are present. Consequently, it would be undesirable to use salinities from sites upstream of Lock 5 Upper as inputs. The model inputs and outputs

used (training / testing set 14) are shown in Table 3.159. The total number of inputs used was 56 and the total number of outputs used was 1 (at lag -27).

Table 3.159: Inputs Used for Forecasting Salinity at Murray Bridge 28 Days in Advance (Training / Testing Set 14)

Input time series	Lags of inputs (days)	Total number of inputs
SMB	1, 3, ..., 11	6
SMN	1, 3, ..., 15	8
SMO	1, 3, ..., 15	8
SWE	1, 2, ..., 5	5
SLO	1, 2, ..., 5	5
SL5U	1, 2, ..., 10	10
FOC	-19, -17, ..., 7	14

3.9.3 Choice of Internal Parameters and Network Geometry

Using the experience gained from Section 3.5, a learning rate of 0.02, a momentum value of 0.6, the hyperbolic tangent transfer function and the quadratic error function were used. In order to increase training speed, an epoch size of 5 was used instead of an epoch size of 16. The network geometry was chosen to be 56-60-20-1 for all tests, as similar network configurations gave good results in Section 3.5.

3.9.4 Investigation of the Generalisation Ability of the Models

In this section, the ability of MANN models to forecast salinity in the River Murray at Murray Bridge 28 days in advance is investigated. Four networks were trained using training / testing set 14, each using different years of data for training and testing. Forecasts for 1988 were obtained in model SMB_M_A_80_14_88, forecasts for 1989 were obtained in model SMB_M_A_80_14_89, forecasts for 1990 were obtained in model SMB_M_A_80_14_90 and forecasts for 1991 were obtained in model SMB_M_A_80_14_91. In each case, the data that were not used for forecasting were used for training. As pointed out in Section 3.5.2.4, the choice of which years are used for training and which year is used for testing can have a significant impact on the generalisation ability of a model. Consequently, by developing several models, each using different years of data for training and testing, a good indication of the generalisation ability of the MANN models for forecasting salinity in the River Murray at Murray Bridge 28 days in advance can be obtained.

Cross-validation was used as the stopping criterion, and the models that produced the best forecasts were retained. The testing interval was chosen to be 1,000. When training the four models, less than 40,000 training samples had to be presented to the networks before a plateau was reached in the RMSE of the test set. The RMSEs at the various stages of learning are given in Appendix E.

The results obtained for the 14 day forecasts (models SMB_M_A_18_10_88, SMB_M_A_18_10_89, SMB_M_A_18_10_90 and SMB_M_A_18_10_91) and the 28 day forecasts (models SMB_M_A_80_14_88, SMB_M_A_80_14_89, SMB_M_A_80_14_90 and SMB_M_A_80_14_91) are compared in Table 3.160 (It should be noted that models SMB_M_A_80_15_XX and SMB_M_A_80_16_XX, the results of which are also shown in Table 3.160, will be discussed in Section 3.9.4.1). It can be seen that:

- As expected, the 14 day forecasts are better than the 28 day forecasts for all four years (i.e. 1988-1991). The RMSE, AAPE and AAE for the 14 day forecasts, averaged over the four years, were 46.1 EC units, 6.4% and 32.4 EC units respectively, while the same values were 61.7 EC units, 8.3% and 42.4 EC units for the 28 day forecasts.
- The variation in RMSE of the forecasts obtained for the different years (i.e. 1988 to 1991) is a lot smaller for the 14 day forecasts than for the 28 day forecasts. This indicates that the generalisation ability of the models is a lot better for the 14 day forecasts. In other words, the forecasting errors obtained for the 14 day forecasts are relatively insensitive to the year used for forecasting. On the other hand, the RMSEs obtained for the 28 day forecasts are very much dependent on which year is withheld during training and subsequently used for testing.
- The 28 day forecasts obtained are still very good, with an average RMSE of 61.7 EC units and an AAPE of 8.3%.

Table 3.160: Best 14 and 28 Day Forecasts Obtained Using Various Training / Testing Sets

Model	Performance measure	Year predicted				Average
		1988	1989	1990	1991	
SMB_M_A_18_10_XX	RMSE (EC units)	44.8	49.3	44.5	45.8	46.1
(14 day forecast)	AAPE (%)	5.3	6.8	7.0	6.5	6.4
	AAE (EC units)	31.0	32.4	32.2	34.0	32.4
SMB_M_A_80_14_XX	RMSE (EC units)	53.8	75.4	49.9	67.6	61.7
(28 day forecast)	AAPE (%)	6.3	10.2	8.4	8.3	8.3
(FOC, 20 days ahead)	AAE (EC units)	38.1	47.4	38.8	45.0	42.4
SMB_M_A_80_15_XX	RMSE (EC units)	57.1	68.4	42.0	70.9	59.6
(28 day forecast)	AAPE (%)	7.0	10.2	7.6	9.4	8.6
(FL7L, 14 days ahead)	AAE (EC units)	45.7	47.2	32.9	51.5	44.3
SMB_M_A_80_16_XX	RMSE (EC units)	77.2	85.3	68.9	80.6	78.0
(28 day forecast)	AAPE (%)	9.5	11.5	10.5	9.6	10.3
(FL7L, 0 days ahead)	AAE (EC units)	60.4	55.2	47.0	56.6	54.8

A comparison of the plots of the best 14 day and 28 day forecasts (Figure 3.159) shows that:

- There is very little lag between the actual major variations in salinity and the 28 day forecasts obtained.
- The lags for the 28 day forecasts are slightly greater than those for the 14 day forecasts.
- The 28 day forecasts of rapid changes in salinity (i.e. sharp peaks and troughs) are not very good. This is due to the fact that most of these occur at times of high flows, and consequently use inputs from the sites furthest upstream. However, as can be seen in Figures 3.16, 3.21, 3.26, 3.31, 3.36, these sharp rises and falls in salinity are not present in the data at these sites.

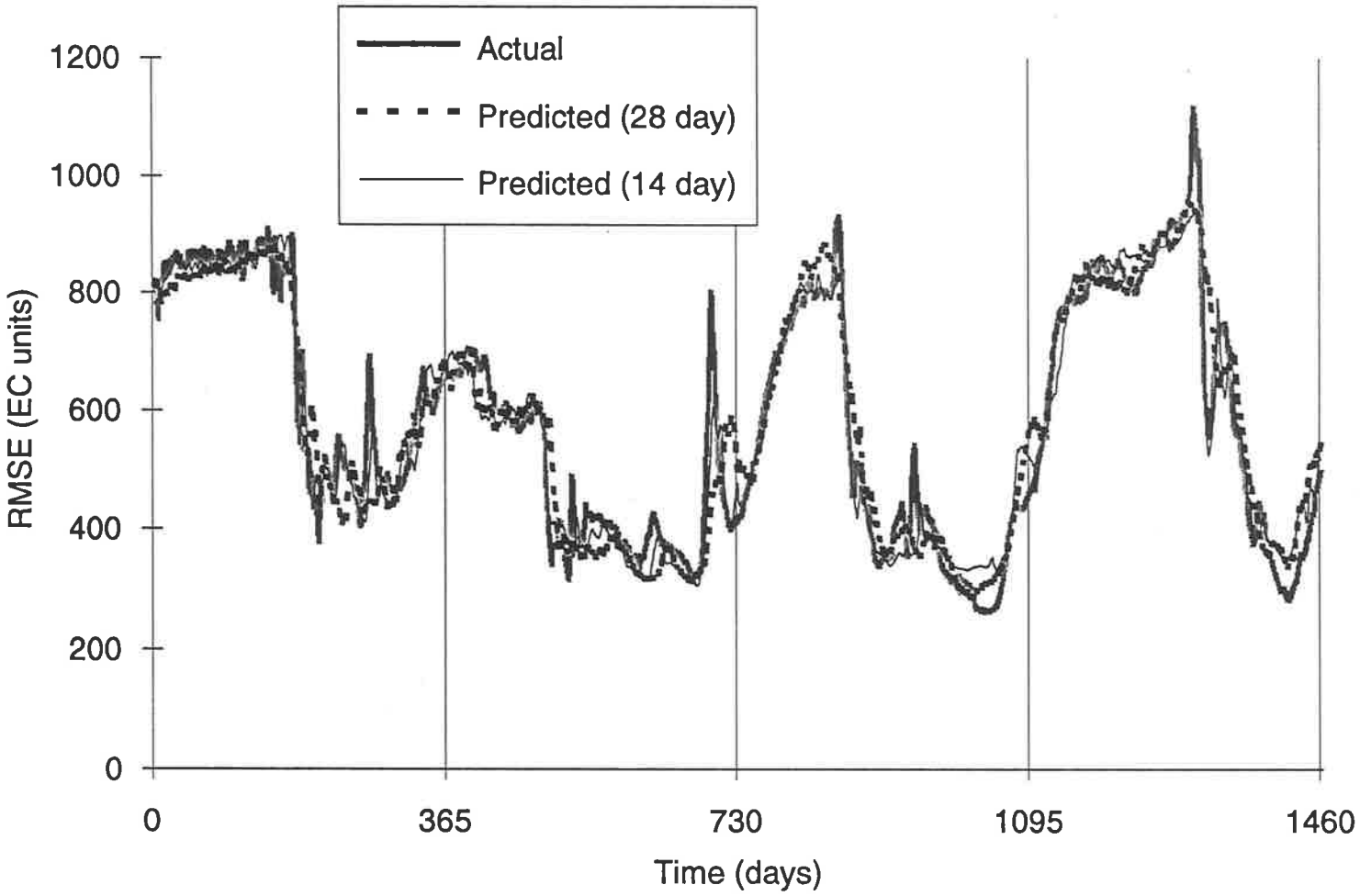


Figure 3.159: Actual and Predicted (14 and 28 Days in Advance) Salinities in the River Murray at Murray Bridge - 1988 to 1991

3.9.4.1 Sensitivity of Models to Flow Inputs

In the above models, it was assumed that flows could be forecast perfectly at Overland Corner up to 20 days in advance. In order to reduce the length of the forecasting period required for the flow inputs, flows from a location further upstream (Lock 7 Lower) were used. Flow forecasts of up to one month in advance are made at this location by the Murray-Darling Basin Commission (Pfeiffer, 1994). No documented records of the accuracy of these forecasts are available, but forecasts at low flows (less than 10,000 ML/day) are usually very accurate, and forecasts of flows above that are generally within ± 10% of the actual values (Pfeiffer, 1994). This order of accuracy is confirmed by trial 14 day flow forecasts carried out by the Murray-Darling Basin Commission at Wentworth (located just downstream of the junction of the Murray and Darling Rivers, Figures 3.1 and 3.2) over a period of 63 days (Murray-Darling Basin Commission, 1990). The trial was conducted during a period of high flows, and the mean of the differences between the historical and predicted values was 1415 ML/day, which is approximately equal to an AAPE of 8.1%.

The above results indicate that it is reasonable to use flow forecasts of up to one month in advance, although forecasts of no more than 14 days would be preferable.

Using Forecast Flows at Lock 7 Lower 14 Days in Advance

The training / testing set used (training / testing set 15) was identical to training / testing set 14 (Appendix C), with the exception that the flows at Overland Corner at lags -19, -17, ..., 7 were replaced with flows at Lock 7 Lower (FL7L) at lags -13, -11, ..., 13 (training / testing set 15, Appendix C). Networks were trained to produce 28 day forecasts of salinity at Murray Bridge for 1988 (model SMB_M_A_80_15_88), 1989 (model SMB_M_A_80_15_89), 1990 (model SMB_M_A_80_15_90) and 1991 (model SMB_M_A_80_15_91).

The testing interval was chosen to be 1,000. When training the four models, less than 40,000 training samples had to be presented to the networks before a plateau was reached in the RMSE of the test set. The RMSEs at the various stages of learning are given in Appendix E.

The results obtained are summarised in Table 3.160. It can be seen that the forecast errors obtained when using flows at Lock 7 Lower 14 days in advance are comparable with those obtained when using flows at Overland Corner 20 days in advance. This indicates that it is worthwhile using flow inputs from sites further upstream, as the same predictive ability of the network can be achieved with shorter flow forecasts.

Using Known Flows at Lock 7 Lower

In order to assess the effect of not using any forecast flows as inputs, the latest known flows at Lock 7 Lower were used as model inputs. The training / testing set used (training / testing set 16) was identical to training / testing set 14 (Appendix C) with the exception that the flows at Overland Corner at lags -19, -17, ..., 7 were replaced with flows at Lock 7 Lower (FL7L) at lags 1, 3, ..., 27 (training / testing set 16, Appendix C). Networks were trained to produce 28 day forecasts of salinity at Murray Bridge for 1988 (model SMB_M_A_80_16_88), 1989 (model SMB_M_A_80_16_89), 1990 (model SMB_M_A_80_16_90) and 1991 (model SMB_M_A_80_16_91).

The testing interval was chosen to be 1,000. When training the four models, less than 40,000 training samples had to be presented to the networks before a plateau was reached in the RMSE of the test set. The RMSEs at the various stages of learning are given in Appendix E.

The results obtained are summarised in Table 3.160. It can be seen that when the known flows at Lock 7 Lower were used as inputs, the forecasts obtained were significantly worse. A comparison of the plots of the 28 day forecasts using flow inputs from Lock 7 Lower 14 days in advance (FL7L_14) and 0 days in advance (FL7L_0) (Figure 3.160) shows an increase in the lag between actual and predicted salinities when the last known flow values are used instead of the predicted ones. It should be noted, however, that using the last known flows is the worst case scenario. Judging from the accuracy of the flow forecasts obtained (Murray-Darling Basin Commission, 1990), one would expect the salinity forecasts obtained, when forecast flows are used, to be very close to the ones obtained when the actual values of flow at Lock 7 Lower, up to 14 days in advance, were used.

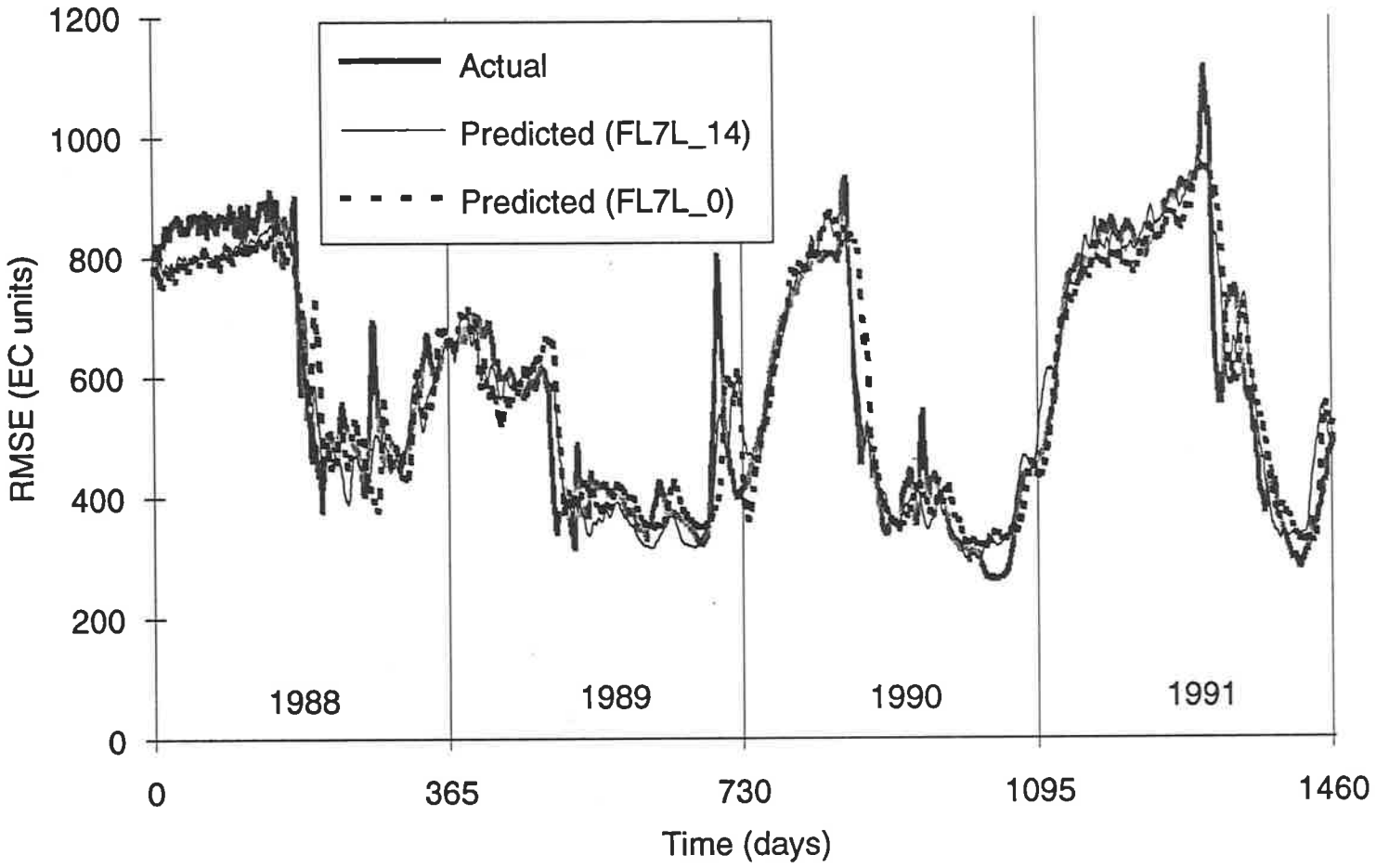


Figure 3.160: Actual and Predicted Salinities (28 Days Ahead) Using Actual (FL7L_0) and Predicted (FL7L_14) Flows at Lock 7 Lower

3.9.5 Real Time Forecasting

A real time forecasting simulation was carried out by using data from 1987 to 1990 for training and data from 1991 for forecasting. In addition, a fixed number of training samples were presented to the network. Initially, a network was trained that used forecasts of flow at Lock 7 Lower up to 14 days in advance as inputs (model SMB_M_A_81_15_91). The effect of using only the latest known inputs of flow at Lock 7 Lower was also investigated (model SMB_M_A_81_16_91). The network parameters and geometries used were the same for both models (Appendix D). A learning rate of 0.002 was used in order to reduce the magnitude of the oscillations in the RMS forecasting error. The other network parameters used include a momentum value of 0.6, an epoch size of 16, the hyperbolic tangent transfer function and the quadratic error function. A network geometry of 56-50-0-1 was used, which is similar to the one used to obtain the 1, 5 and 14 day real time forecasts (i.e. 51-45-0-1, Section 3.5.4). In order to ensure a local minimum in the error surface had been reached when training was stopped, 150,000 training samples were presented to both networks.

The results obtained are summarised in Table 3.161. It can be seen that the RMSE of the forecasts obtained when forecast flows were used as inputs is lower than that of the forecasts obtained when only the latest known values of flows were used. This suggests that flow inputs assist in forecasting sharp changes in salinity. This is confirmed by the fact that the AAPEs for both models are very similar. It is interesting to note that the 28 day forecasts obtained using the multivariate ANN model had a lower RMSE than the 14 day forecasts obtained using the VARIMA models.

Table 3.161: Real Time Forecasts Obtained 28 Days in Advance

Model	Inputs	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_M_A_81_15_91	With forecast flows	74.5	10.0	53.3
SMB_M_A_81_16_91	Without forecast flows	78.3	9.8	55.4

A plot of the 28 day real time forecast obtained using model SMB_M_A_81_15_91 is shown in Figure 3.161. By comparing Figure 3.131 (14 day forecast) with Figure 3.161 (28 day forecast), it can be seen that the shift between the actual and predicted salinities is increased for the 28 day forecast. However, most major variations in salinity are still forecast without large lags, although some of the shifts between actual and predicted salinities are starting to become significant.

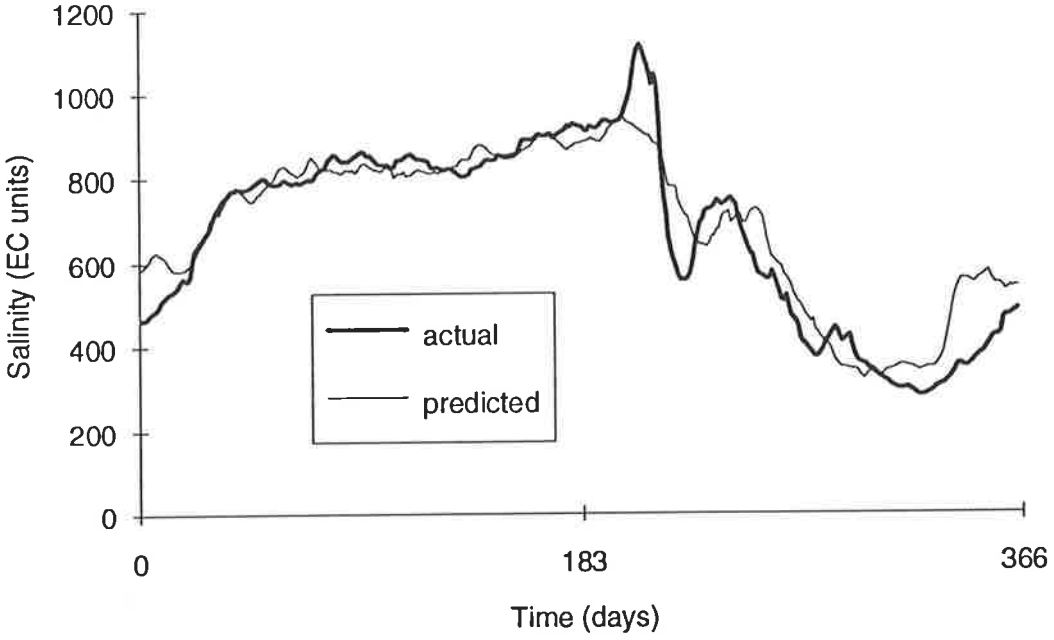


Figure 3.161: Real Time Forecast of Salinity in the River Murray at Murray Bridge Obtained Using Model SMB_M_A_81_15_91 - 1991 - 28 Days in Advance

A comparison of the RMSEs for the 1, 5, 14 and 28 day forecasts for 1991 is shown in Figure 3.162. It can be seen that the relationship between RMS forecasting error and forecasting length is almost linear. Figure 3.162 also indicates that the inclusion of future values of flow in the input set becomes more important with increasing forecasting period. This is because for long forecasting periods (e.g. 28 days), the salinity travel times from the most upstream site (i.e. Lock 5 Upper) to Murray Bridge is less than the forecasting period, so that variations in flow give the only indication about variations in salinity.

This is confirmed by inspecting typical plots of the relative significance of the various model inputs at different forecasting periods (Figures 3.163 to 3.166). It can be seen that flow inputs become increasingly important as the forecasting period increases. The same holds for salinity inputs from sites further upstream. This is not surprising, as the salinity travel time should ideally be equal to the forecasting period. This is because salinity travel times increase with increasing distance upstream.

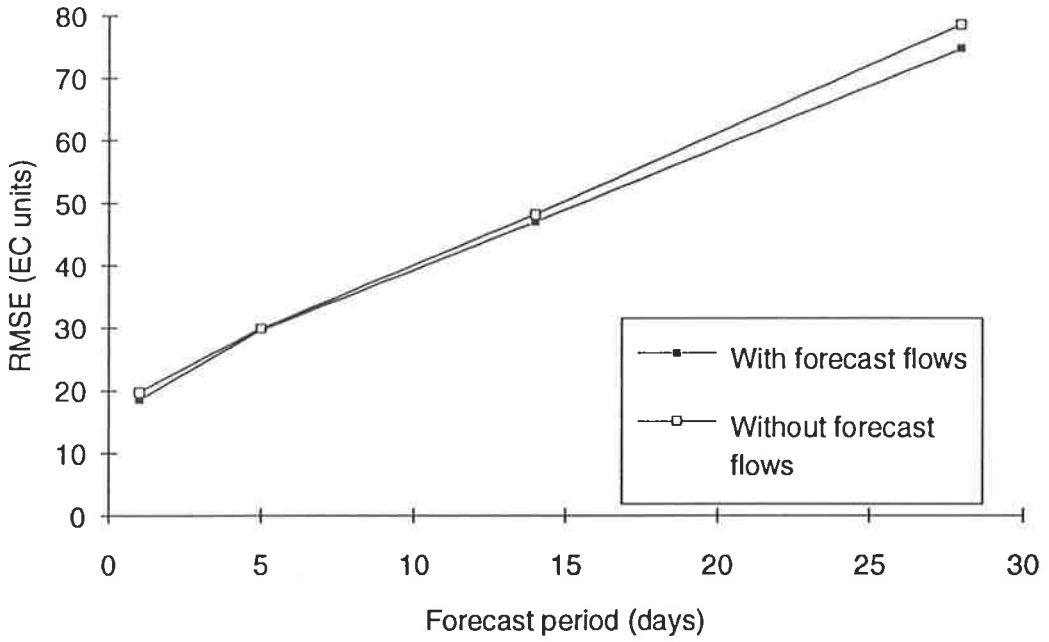


Figure 3.162: Change in the RMSE of the Real Time Forecasts of Salinity in the River Murray at Murray Bridge with Forecasting Period (1991)

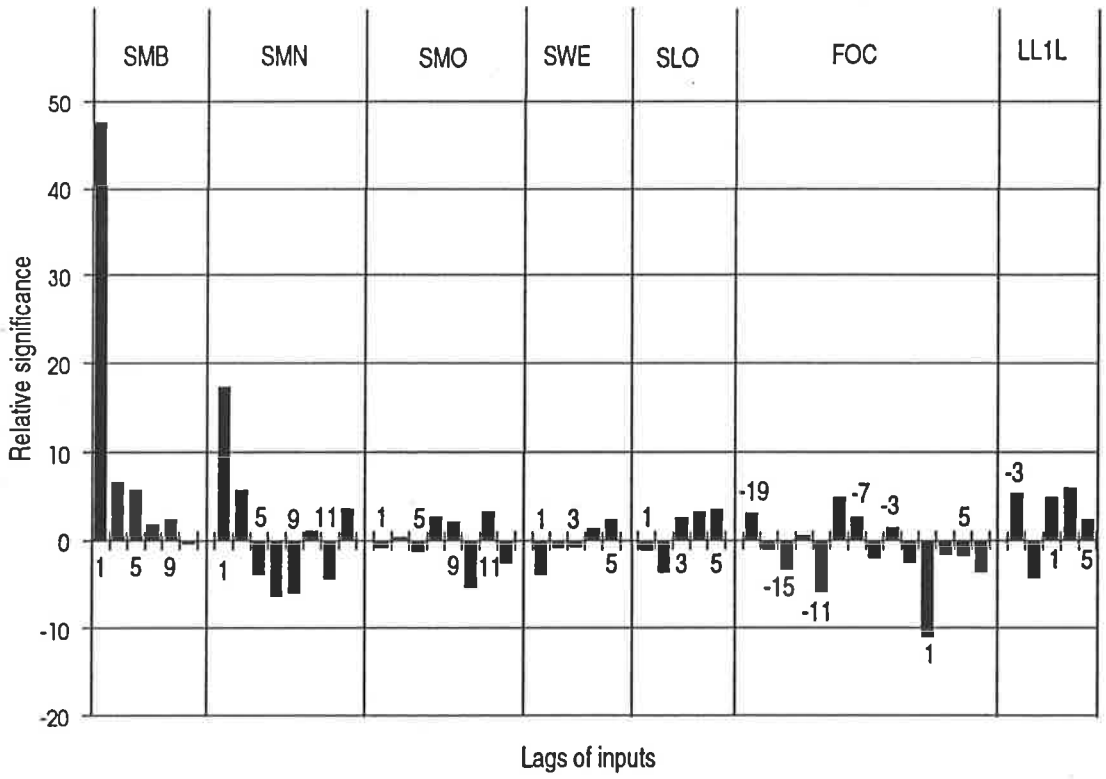


Figure 3.163: Typical Plot of the Relative Significance of the Model Inputs at Various Lags - 1 Day Forecast

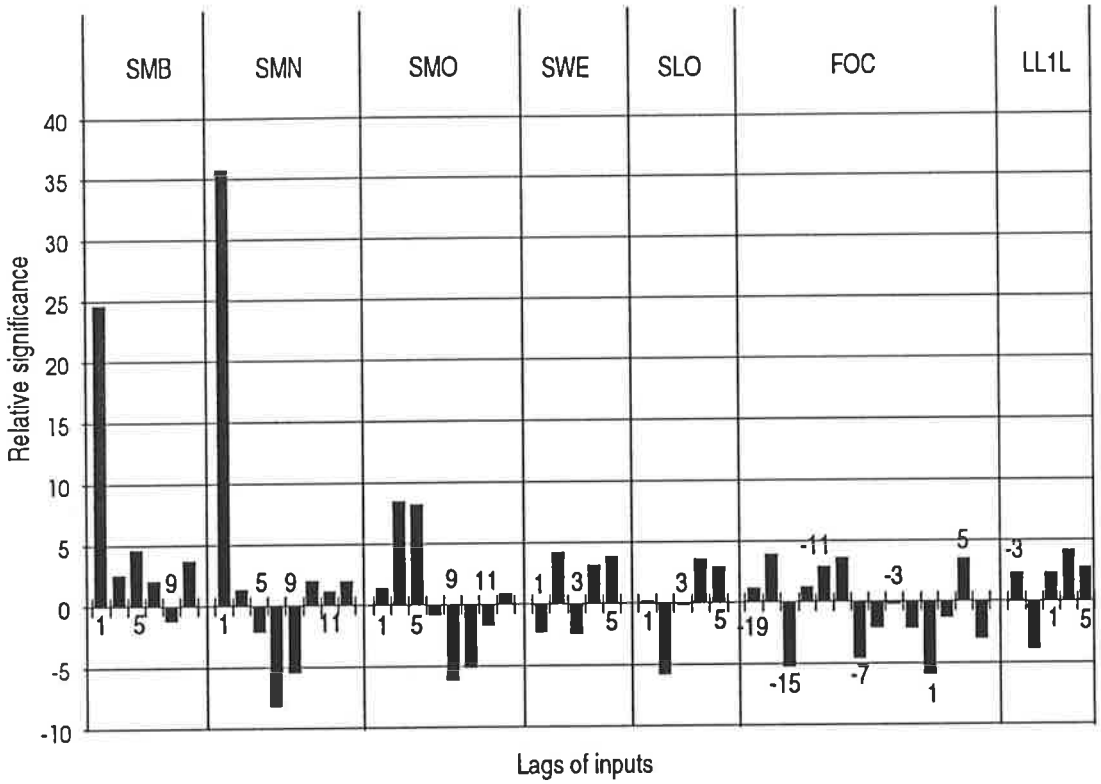


Figure 3.164: Typical Plot of the Relative Significance of the Model Inputs at Various Lags - 5 Day Forecast

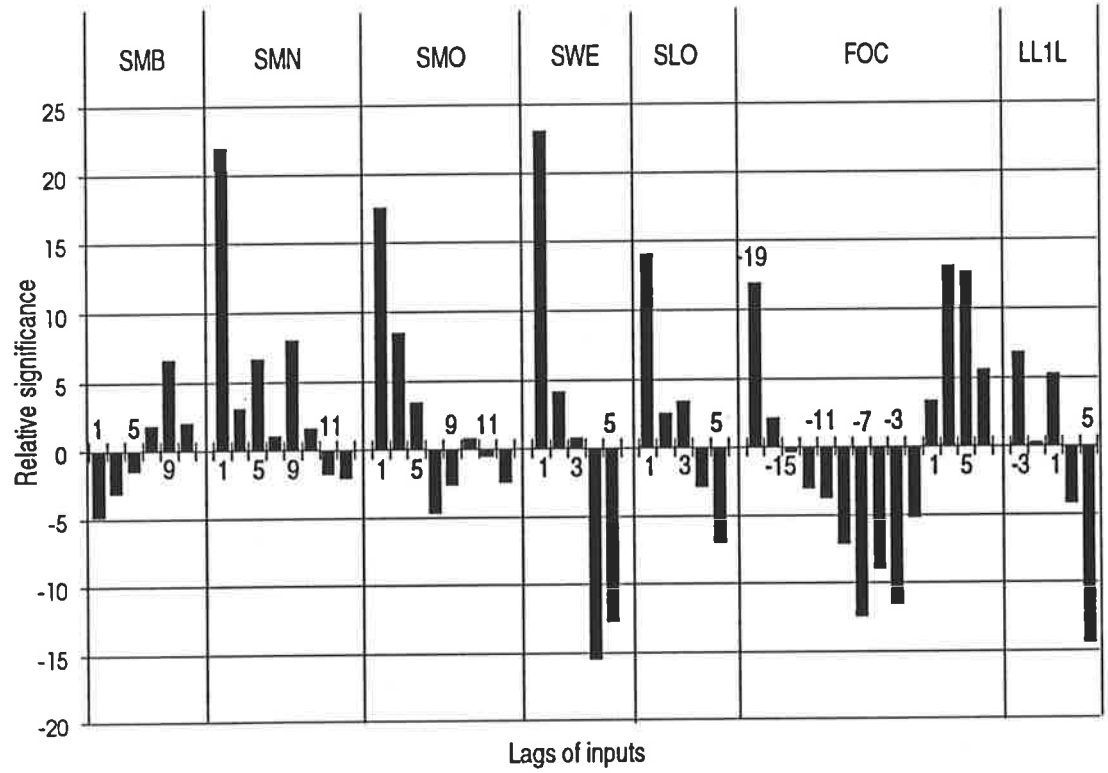


Figure 3.165: Typical Plot of the Relative Significance of the Model Inputs at Various Lags - 14 Day Forecast

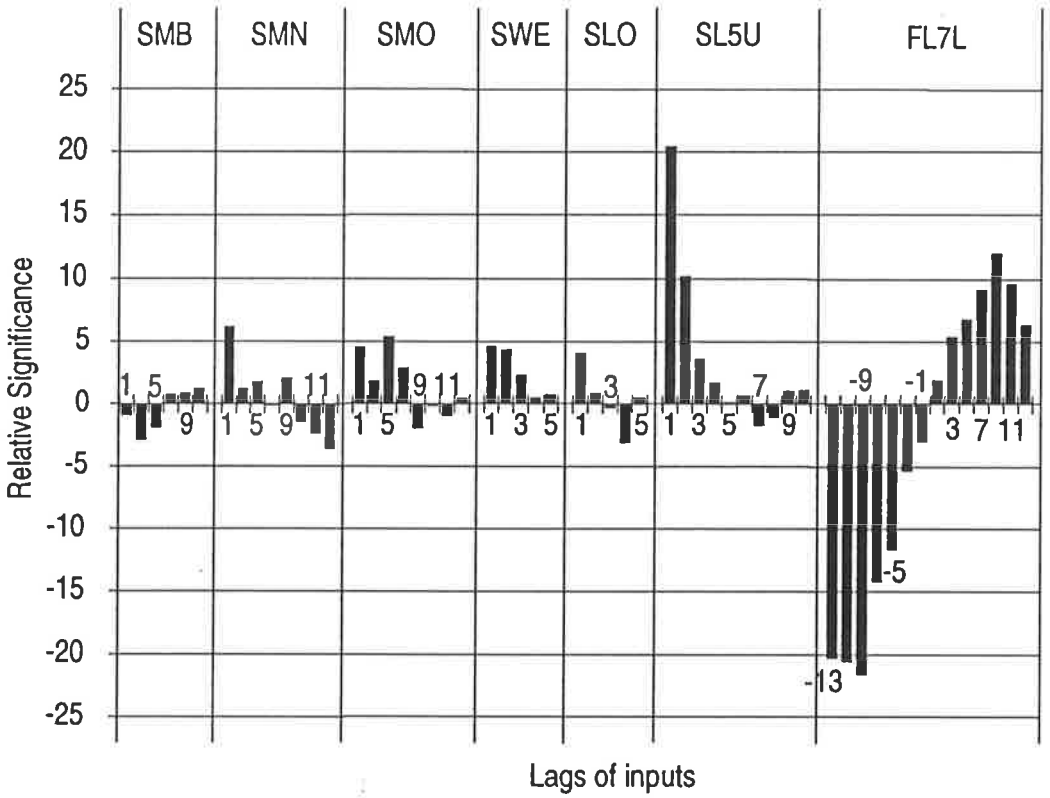


Figure 3.166: Typical Plot of the Relative Significance of the Model Inputs at Various Lags - 28 Day Forecast

3.10 Methods for Determining Inputs to Multivariate Neural Network Models

In Section 3.5, information about the underlying physical processes responsible for high levels of salinity (i.e. salinity transport and inflows of saline groundwater) were used in conjunction with extensive sensitivity analyses to determine appropriate inputs for the multivariate ANN model. However, there are a number of problems associated with this procedure, including:

- It is not a generalised procedure, as it can only be used if some *a priori* knowledge about the data is available.
- The sensitivity analyses are time consuming, as the number of initial inputs is generally large, resulting in large networks. This leads to a number of problems as discussed in Sections 2.3.2 and 3.8.6.

As discussed in Section 2.3.2, Lachtermacher (1993) and Lachtermacher and Fuller (1994) developed a hybrid methodology for determining the inputs for simple univariate

ANN models. However, no methodology exists for determining inputs to multivariate ANN models.

In this Section, it is proposed that the method of Haugh and Box (1977) can be used to determine the inputs to multivariate ANN models. In addition, a new method for determining the inputs for multivariate ANN models is introduced. This method involves the development of simple univariate and bivariate ANN models to establish relationships between the output time series and past values of each of the input time series. The strength of these relationships at various lags can then be determined by means of a sensitivity analysis, and the inputs that have a significant effect on the outputs chosen as inputs to the multivariate model.

The method of Haugh and Box (1977) and the neural network based approach were applied to the case study of forecasting salinity in the River Murray at Murray Bridge 14 days in advance. The resulting model inputs were compared with those obtained in Section 3.5.4 (training / testing set 13). All three sets of inputs were then used to develop ANN models for forecasting salinity in the River Murray at Murray Bridge, in order to assess the adequacy of the three methods for determining inputs to multivariate ANN models.

As a result of the strong relationship between the flow and level data (Section 3.7.6.2), it was decided to only include flow and salinity inputs. The input time series considered include salinities at Murray Bridge (SMB), Mannum (SMN), Morgan (SMO), Waikerie (SWE) and Loxton (SLO), as well as flow at Lock 1 Lower (FL1L).

3.10.1 Determination of Model Inputs Using the Method of Haugh and Box (Method 1)

The method of Haugh and Box (1977) is described in detail in Section 2.2.13.5. It uses cross-correlation analysis to determine the strength of the relationship between the output time series and the input time series at various lags. However, when using cross-correlation analysis, each of the time series needs to be "prewhitened". This is done by fitting a univariate ARMA type model to each time series and calculating the difference between the historical data and the values predicted by the models, which are called the model residuals. The CCF between the residuals of the output time series and the residuals of each of the input time series is then calculated.

The method of Haugh and Box has already been applied to the case study of forecasting salinity in the River Murray at Murray Bridge (Section 3.7.6.2). The model inputs

chosen using this method (see Table 3.123) are summarised in Table 3.162. It should be noted that for each time series, all values from lag 1, up to and including the maximum significant lag, were chosen as model inputs.

Table 3.162: Model Inputs Obtained Using the Method of Haugh and Box (Method 1)

Time series	Lags of inputs
SMB	1, 2, 365, 366
SMN	1, 2, 3, 4
SMO	1, 2, 3, 4, 5, 6, 7
SWE	1, 2, ,3, 4, 5, 6, 7, 8, 9, 10, 11
SLO	1, 2, ,3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
FL1L	1, 2, ,3, 4, 5, 6, 7, 8, 9, 10

3.10.2 Determination of Model Inputs Using the Neural Network Based Approach (Method 2)

An outline of this procedure is shown in Figure 3.167.

3.10.2.1 Inspection of Plots of the Time Series

The plots of the input time series should be inspected for reasons discussed in Section 2.2.6.1. Plots and discussions of the input and output time series for the case study considered are given in Section 3.3.3.1.

3.10.2.2 Development of Univariate and Bivariate ANN Models

If the number of time series is N' (i.e. z_1, z_2, z_3, ..., $z_{N'}$), and the output time series is z_1, N' neural network models have to be developed. Network 1 predicts z_1 using its own past values as inputs, network 2 predicts z_1 using past values from z_2 as inputs, network 3 predicts z_1 using past values from z_3 as inputs and so on. The following steps form part of the model development procedure:

(i) Choice of lags of inputs and outputs:
For each network, only one input time series is used. A maximum lag (k_{max}) is chosen, and values at lags 1, 2, 3, ..., k_{max} are used as model inputs. If *a priori* knowledge about the relationship between the input and output time series is available, k_{max} is chosen so that the lags of the input time series that exceed k_{max} are not suspected to have any significant effect on the output time series. If no *a priori* knowledge is

available, k_{max} has to be chosen arbitrarily. The number and lags of the outputs depends on the forecasting period(s) required.

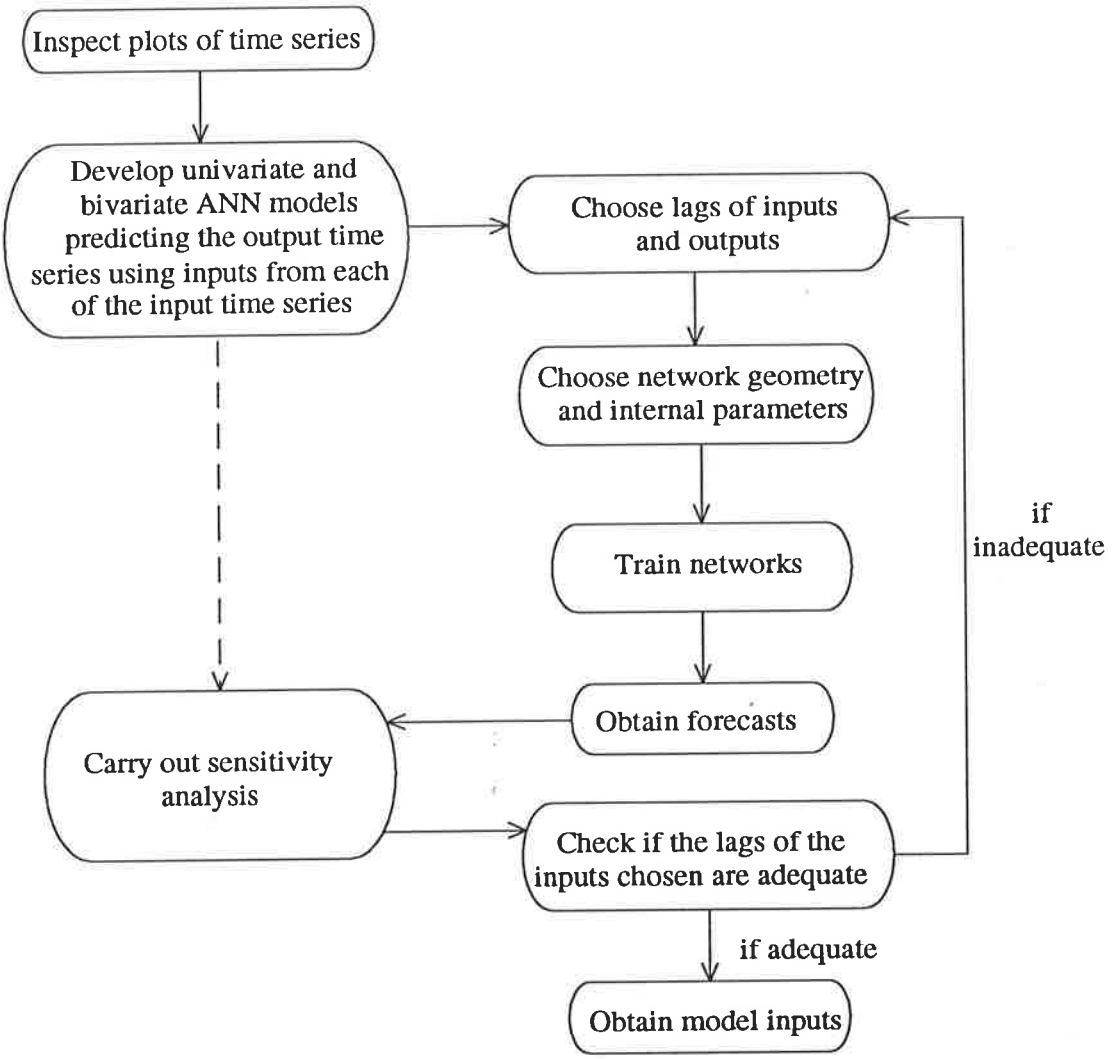


Figure 3.167: Outline of Neural Network Based Approach (Method 2)

In the case study considered, the number of input time series is six (i.e. N' = 6). A summary of the inputs and outputs chosen for each of the six ANN models developed is given in Table 3.163. The value of k_{max} was chosen to be 20 for the model using salinities at Murray Bridge as inputs (model SMB_U_A_82_12_88). This is conservative, as one would not expect future values of salinity at Murray Bridge to be strongly related to their own past values up to lags of that magnitude. In addition to the inputs at lags 1 to 20, inputs in the vicinity of one seasonal lag (i.e. 354 to 376) were included, to obtain a fair comparison between the inputs obtained using this method and the method of Haugh and Box.

Table 3.163: Details of the Univariate and Bivariate Models Trained for Input Identification Purposes

Model	Output time series	Input time series	Lags of inputs (days)
SMB_U_A_82_17_88	SMB	SMB	1, 2, ..., 20, 354, 355, ..., 376
SMB_M_A_83_18_88	SMB	SMN	1, 2, ..., 30
SMB_M_A_83_19_88	SMB	SMO	1, 2, ..., 30
SMB_M_A_83_20_88	SMB	SWE	1, 2, ..., 30
SMB_M_A_83_21_88	SMB	SLO	1, 2, ..., 30
SMB_M_A_84_22_88	SMB	FL1L	1, 2, ..., 30

Taking into account the salinity travel times from upstream locations to Murray Bridge, and the fact that the models are trained for a forecasting period of 14 days, a value of $k_{max} = 30$ was chosen for the models using upstream salinities as inputs. For the model using flows from Lock 1 Lower as inputs (model SMB_M_A_84_22_88), an arbitrary choice of $k_{max} = 30$ was made. For all models, the lag of the output was chosen to be -13, as the desired forecasting period is 14 days.

(ii) Choice of network geometry and internal parameters:

As indicated by the results obtained in Section 3.5.3, there are a wide range of network geometries and internal parameters for which network performance is only slightly affected. It should be noted that at this stage, it is not crucial that the best possible forecasts are obtained, as the aim of the procedure is to obtain the dominant network inputs. The dominant inputs will generally be the same, regardless of which geometry and internal parameters have been chosen, unless the network gets stuck in an undesirable region of the weight space during training.

Using the experience gained from Section 3.5.3, the hyperbolic tangent transfer function, the quadratic error function, an epoch size of 16, a momentum of 0.6 and a learning rate of 0.1 were used for all models. A learning rate of 0.1 resulted in divergent behaviour for the model using inputs of flows at Lock 1 Lower. Consequently, the learning rate was reduced to 0.01. The number of nodes used in the first hidden layer was 45 and the number of nodes used in the second hidden layer was 15 for all models.

(iii) Training:

The stopping criterion used was cross-validation. The testing interval was chosen to be 5,000. In each case, a local minimum in the error surface was reached prior to the

presentation of 50,000 training samples. The RMSEs at the various stages of learning are given in Appendix E.

(iv) Forecasting:

This step is optional. However, by assessing the forecasting accuracy of each of the models using performance measures such as those discussed in Section 2.1.8.2, the relative significance of each of the input time series in predicting the output time series can be obtained. Plots of the actual and predicted time series can also provide valuable information.

The best results obtained using the models with the various input time series are summarised in Table 3.164. It can be seen that for the models using salinity values as inputs, there is a conflict between:

- Increased performance when using inputs from sites further upstream as a result of longer salinity travel times, which enable longer term forecasts with smaller lags.
- Decreased performance when using inputs from sites further upstream, due to higher levels of salinity and the presence of additional salinity peaks at downstream sites as a result of saline accessions.

Table 3.164: Best Results Obtained Using the Various Univariate and Bivariate Models - 14 Day Forecasts

Model	Input time series	RMSE (EC units)
SMB_U_A_82_17_88	SMB	87.5
SMB_M_A_83_18_88	SMN	67.2
SMB_M_A_83_19_88	SMO	54.9
SMB_M_A_83_20_88	SWE	59.9
SMB_M_A_83_21_88	SLO	64.9
SMB_M_A_84_22_88	FL1L	87.9

Plots of the 14 day forecasts obtained using salinities at Murray Bridge as inputs (Figure 3.168) and those obtained using salinities at Waikerie as inputs (Figure 3.169) demonstrate the reduction in the shift between the actual and predicted values on the rising and falling limbs of the flood hydrograph when salinities from the site further upstream (i.e. Waikerie) are used as inputs.

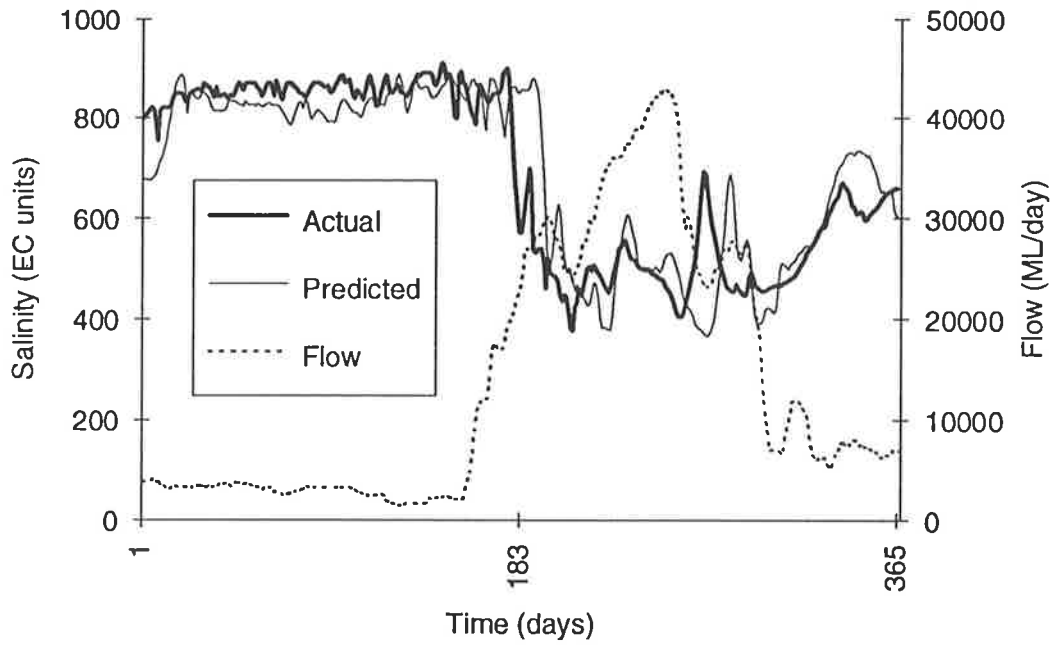


Figure 3.168: Actual and Predicted Salinities - Model SMB_U_A_82_17_88 (i.e. Model Using Salinities at Murray Bridge as Inputs) - 14 Days in Advance

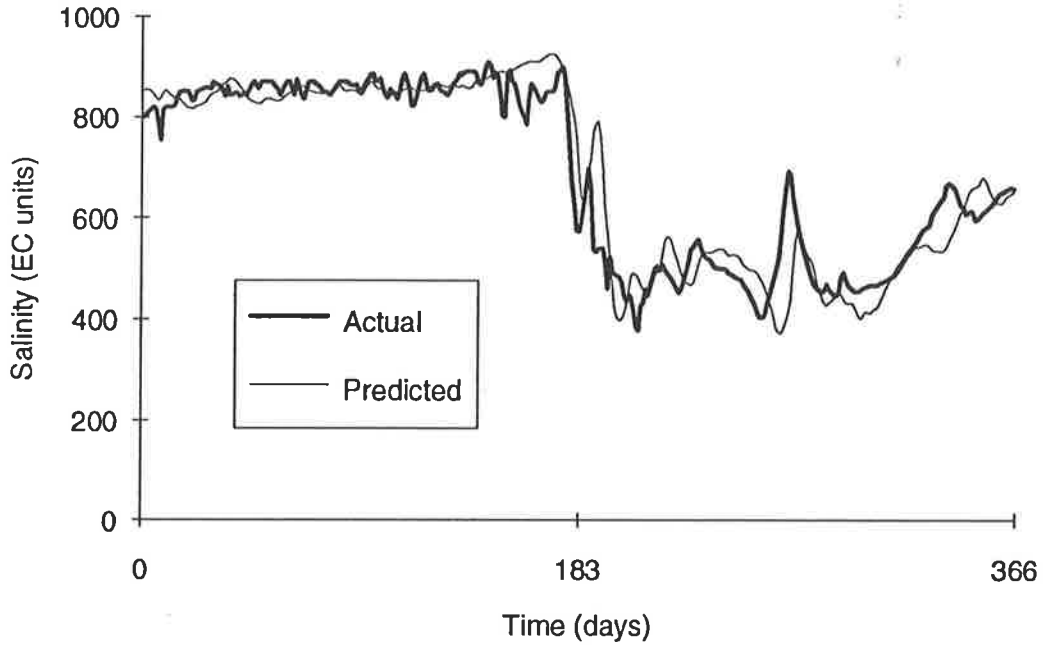


Figure 3.169: Actual and Predicted Salinities - Model SMB_M_A_83_20_88 (i.e. Model Using Salinities at Waikerie as Inputs) - 14 Days in Advance

A plot of the 14 day forecasts obtained when flows at Lock 1 Lower were used as inputs is shown in Figure 3.170. It can be seen that using flows at Lock 1 Lower as inputs only enables the prediction of the major variations in salinity at Murray Bridge,

without capturing any of the detailed variations. This is reflected in the high RMSE obtained for model SMB_M_A_84_22_88 (Table 3.164).

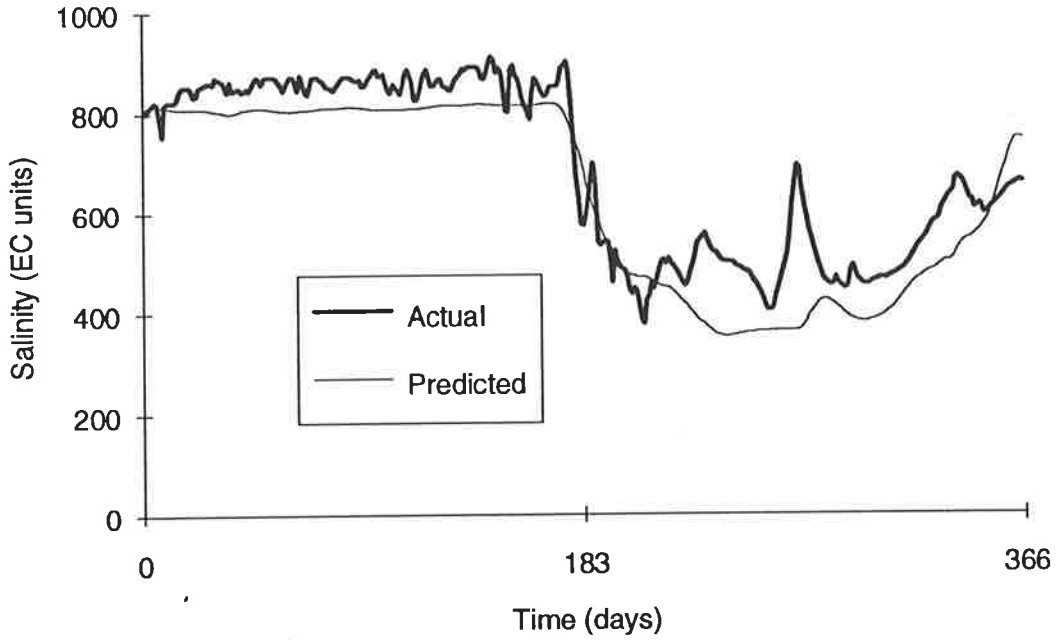


Figure 3.170: Actual and Predicted Salinities - Model SMB_M_A_84_22_88 (i.e. Model Using Flows at Lock 1 Lower as Inputs) - 14 Days in Advance

3.10.2.3 Performance of Sensitivity Analyses

In this step, the strength of the relationship between the model outputs(s) and the model inputs is determined for the models developed in Section 3.10.2.2 with the aid of sensitivity analyses. By examining plots of the relative significance of the model inputs at different lags for each of the models, appropriate inputs for the multivariate model can be chosen. The choice of which level of significance warrants the inclusion of a particular input is somewhat arbitrary and requires some degree of judgement.

The plots of the relative significance of the model inputs can also be used to check if the value of k_{max} chosen (Section 3.10.2.2) was large enough. If the relative significance of the input at lag k_{max} is high, the value of k_{max} is too small. In this case, a larger value of k_{max} has to be chosen, the network geometry and internal parameters have to be adjusted and training and the sensitivity analysis have to be repeated.

In the case study considered, the sensitivity analyses were carried out with the aid of the software's (NeuralWare Professional II/Plus) "Explain" function (see Section 3.3.4.1).

The relative significance of the inputs was assessed separately for high and low flow conditions, as the significance of the various inputs is flow dependent.

A typical plot of the relative significance of the inputs during high flow and low flow conditions for model SMB_U_A_82_17_88 (i.e. the model using salinities at Murray Bridge as inputs) is shown in Figure 3.171 . It can be seen that there are no significant inputs at high lags, indicating that the value of k_{max} chosen was large enough. It is interesting to note that there are no significant inputs in the vicinity of one seasonal lag, which is in contrast to the results obtained using the method of Haugh and Box (Table 3.162). By inspecting Figure 3.171, inputs of salinity at Murray Bridge at lags 1 and 2 were considered significant, and hence chosen as inputs for the multivariate model.

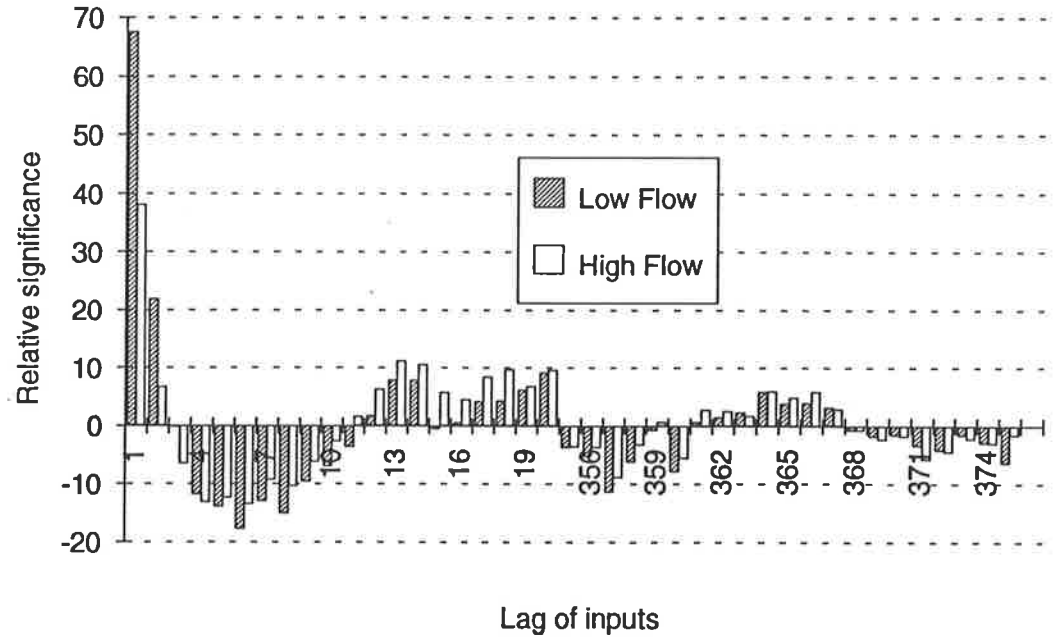


Figure 3.171: Relative Significance of Inputs - Model SMB_U_A_82_17_88 (i.e. Model Using Salinities at Murray Bridge as Inputs) - 14 Days in Advance

As the distance between Murray Bridge and the location of the input time series increases, the plots of the relative significance of the inputs (a typical plot is shown in Figure 3.172) exhibit the following trends:

- At small lags, the inputs during high flow conditions become increasingly significant when compared with those during low flow conditions.
- The number of significant lags increases.

The above observations are in agreement with what one would expect. During low flow conditions, salinity travel times from the sites closer to Murray Bridge (e.g. Mannum)

exceed 14 days and provide sufficient information to produce adequate 14 day forecasts. Consequently, salinity inputs from sites further upstream are not needed during low flow conditions. However, during high flow conditions, salinities from sites further upstream are required to produce adequate 14 day forecasts, as salinity travel times from the sites closer to Murray Bridge are less than 14 days.

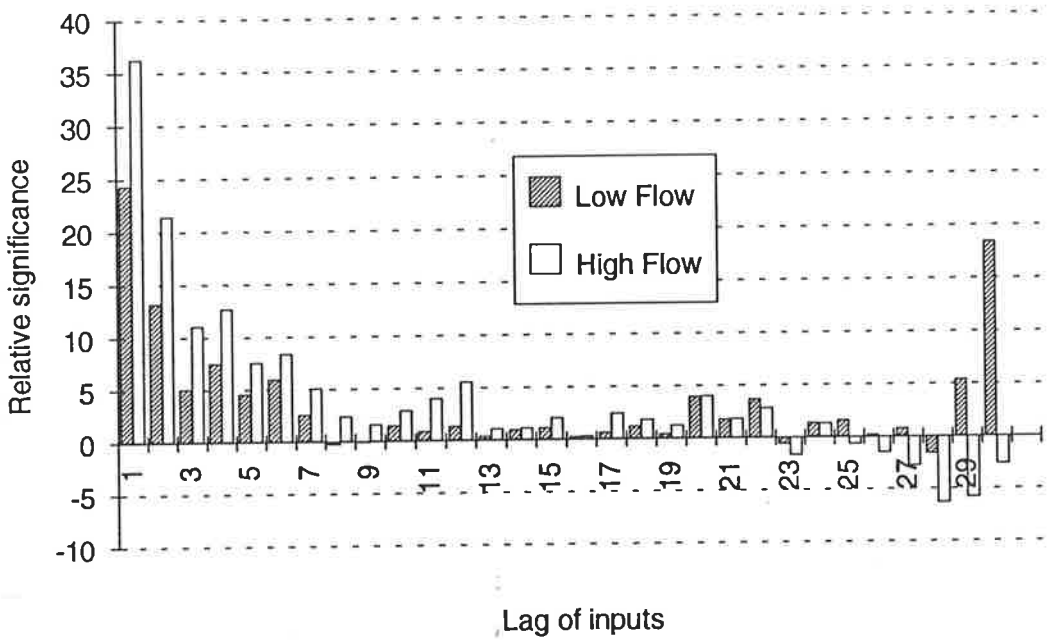


Figure 3.172: Relative Significance of Inputs - Model SMB_M_A_83_21_88 (i.e. Model Using Salinities at Loxton as Inputs) - 14 Days in Advance

The lags of the inputs that were found to be significant for the models using salinity at Mannum, Morgan, Waikerie and Loxton as inputs are summarised in Table 3.165.

Table 3.165: Model Inputs Obtained Using the Method Based on a Neural Network Approach (Method 2)

Time series	Lags of inputs
SMB	1, 2,
SMN	1, 2
SMO	1, 2
SWE	1, 2, ,3, 4
SLO	1, 2, ,3, 4, 5, 6, 7
FL1L	1, 2, ,3, 4, 5, 6, 7, 8

Figure 3.172 indicates that there is a significant input at lag 30 during low flow conditions, suggesting that the salinity travel times from Loxton to Murray Bridge exceed 30 days at times of low flow. However, for sites upstream of Morgan, the significance of the inputs during low flow conditions may be ignored. This is because salinities from Murray Bridge, Mannum and Morgan contribute to the prediction of salinity at Murray Bridge at times of low flow, whereas salinities from sites further upstream, including Waikerie and Loxton, mainly contribute to the prediction of salinity at Murray Bridge at times of high flow. During high flow conditions, there are no significant inputs at high lags, indicating that the value of k_{max} chosen was appropriate. Using the above criteria, the value of k_{max} chosen was found to be adequate for the remaining models using salinity inputs (models SMB_M_A_83_18_88, SMB_M_A_83_19_88 and SMB_M_A_83_20_88).

A typical plot of the relative significance of the inputs during high flow and low flow conditions for model SMB_U_A_84_22_88 (i.e. the model using Flows at Lock 1 Lower as inputs) is shown in Figure 3.173. It can be seen that there are significant inputs at high lags during low flow and high flow conditions, indicating that the value of k_{max} chosen was not large enough. Consequently, the training and sensitivity analysis steps were repeated using k_{max} = 50 (model SMB_M_A_85_23_88). The internal parameters used were the same as those used for model SMB_M_A_84_22_88. The network geometry used was 50-60-20-1.

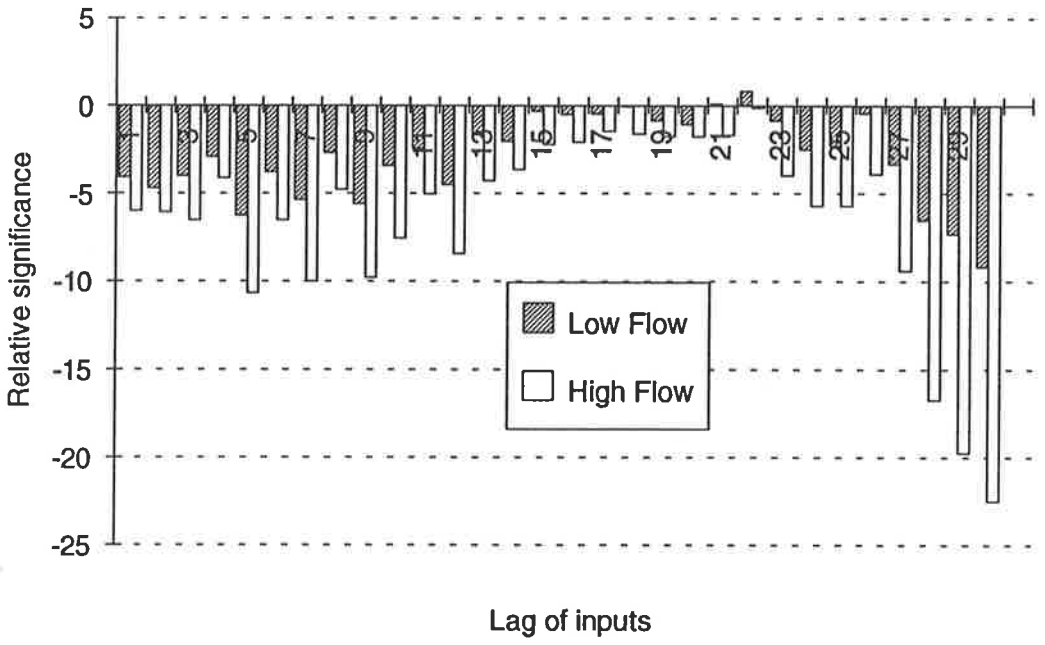


Figure 3.173: Relative Significance of Inputs - Model SMB_M_A_84_22_88 (i.e. Model Using Flows at Lock 1 Lower as Inputs) - 14 Days in Advance

A typical plot of the relative significance of the inputs during high flow and low flow conditions for model SMB_M_A_85_23_88 is shown in Figure 3.174. It can be seen that there are no significant values at high lags, indicating that the lags of the inputs chosen were adequate. It should be noted that the input at lag 50 was not considered to be significant, as it is isolated and less significant than the inputs at lags 1 to 7. It can also be seen that there is a much stronger relationship between flow at Lock 1 Lower and salinity at Murray Bridge at times of high flow.

Figure 3.174 indicates that inputs at lags 1 to 8 appear to be significant for producing 14 day forecasts. Consequently, values of flow at Lock 1 Lower at lags 1 to 8 were chosen as inputs for the multivariate model (Table 3.165).

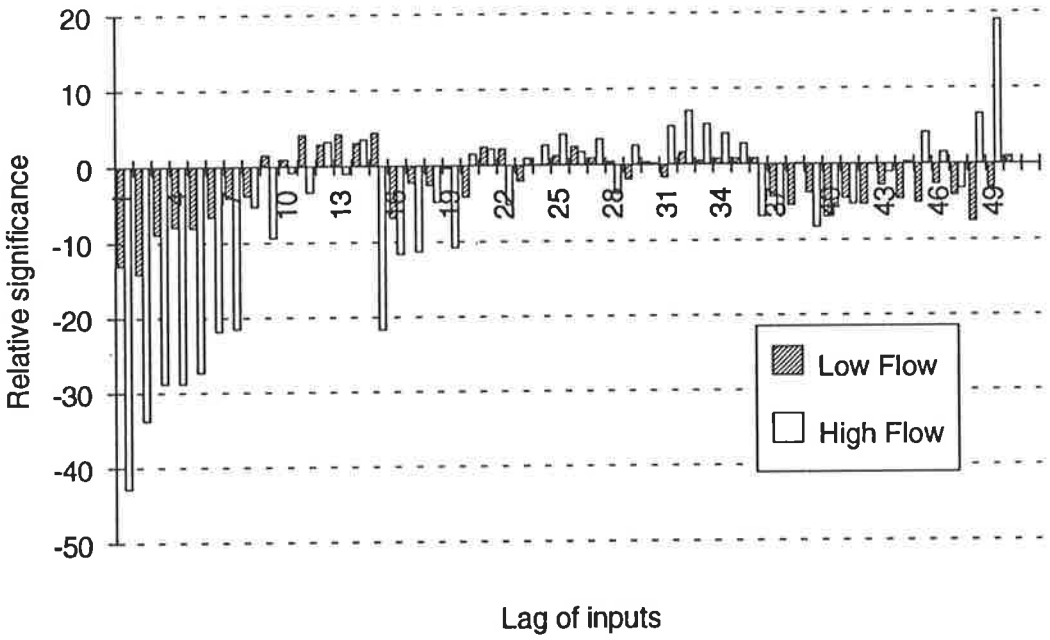


Figure 3.174: Relative Significance of Inputs - Model SMB_M_A_85_23_88 (i.e. Model Using Flows at Lock 1 Lower as Inputs) - 14 Days in Advance

3.10.3 Summary of Network Inputs Chosen

A summary of the network inputs chosen using the method of Haugh and Box (method 1), the neural network based approach (method 2) and the empirical method utilising *a priori* knowledge about the underlying processes causing high salinities (method 3) is given in Table 3.166. It should be noted that it was decided:

- To omit the inputs of salinity at Murray Bridge at lags 365 and 366 identified as important using the method of Haugh and Box, in order to use the same length of data for training the networks using all three input sets.
- To make a number of changes to the inputs obtained using method 3 (training / testing set 13) for the sake of consistency, including replacing the flows at Overland Corner with flows at Lock 1 Lower and omitting the levels at Lock 1 Upper.

Table 3.166: Comparison of Inputs Obtained Using the Various Methods

Input	Method	Lag														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
SMB	1	■	■													
	2	■	■													
	3	■		■		■		■		■		■				
SMN	1	■	■	■	■											
	2	■	■													
	3	■		■		■		■		■		■		■		■
SMO	1	■	■	■	■	■	■	■								
	2	■	■													
	3	■		■		■		■		■		■		■		■
SWE	1	■	■	■	■	■	■	■	■	■	■	■				
	2	■	■	■	■											
	3	■	■	■	■	■										
SLO	1	■	■	■	■	■	■	■	■	■	■	■	■	■		
	2	■	■	■	■	■	■	■								
	3	■	■	■	■	■										
FL1L	1	■	■	■	■	■	■	■	■	■	■					
	2	■	■	■	■	■	■	■	■							
	3	■		■		■		■								

A comparison of the number of salinity, flow and total inputs obtained using the three methods is given in Table 3.167.

Table 3.167: Comparison of the Number of Inputs Obtained Using the Various Methods

	Method 1	Method 2	Method 3
Number of salinity inputs	37	17	32
Number of flow inputs	10	8	4
Total number of inputs	47	25	36

The following points should be noted about Tables 3.166 and 3.167:

- Generally, there is good agreement between the methods.
- Fewer salinity inputs were obtained using method 1 than method 2, as method 2 identifies the critical inputs for a 14 day forecast, whereas method 1 can only identify the critical inputs for a 1 day forecast.
- The number of salinity inputs obtained using method 3 was able to be reduced by taking into account some of the underlying physical principles, resulting in salinity values at alternate days being used from Murray Bridge, Mannum and Morgan. The reason for this is that salinity values from the above locations were assumed to be important at times of low flow, during which changes in salinity with time are slow.
- The lowest number of total inputs was obtained using method 2 and the highest number of total inputs was obtained using method 1.

3.10.4 Real Time Forecasting

The inputs obtained using method 1 (input / output set 24), method 2 (input / output set 25) and method 3 (input / output set 26) were used to obtain real time forecasts for 1991. For each training / testing set, three network geometries were tried. The network geometries used in conjunction with training / testing set 24 include 47-15-0-1 (model SMB_M_A_86_24_91), 47-20-0-1 (model SMB_M_A_87_24_91) and 47-35-0-1 (model SMB_M_A_88_24_91). The network geometries used in conjunction with training / testing set 25 include 25-5-0-1 (model SMB_M_A_89_25_91), 25-15-0-1 (model SMB_M_A_90_25_91) and 25-30-0-1 (model SMB_M_A_91_25_91). The network geometries used in conjunction with training / testing set 26 include 36-15-0-1 (model SMB_M_A_92_26_91), 36-20-0-1 (model SMB_M_A_93_26_91) and 36-35-0-1 (model SMB_M_A_94_26_91). A learning rate of 0.02, a momentum value of 0.6, an epoch size of 16, the hyperbolic tangent transfer function and the quadratic error function were used for all models. This is identical to the internal parameters used in the real time forecasting simulation carried out in Section 3.5.4 (model

SMB_M_A_79_13_91). As was the case in Section 3.5.4, the number of training samples presented to the models was 100,000.

The results obtained are shown in Table 3.168. It can be seen that the generalisation ability of the various models is comparative, regardless of which method was used to obtain the network inputs. However, the forecasts obtained when training / testing sets 24 and 25 were used were slightly better than those obtained with training / testing set 26. The network geometry chosen had some impact on the results obtained. However, the variation in the forecasting errors obtained using different network geometries could be reduced by decreasing the learning rate, although the time taken for training would be increased.

Table 3.168: Real Time Forecasting Errors for Models Using Inputs Obtained by Methods 1 (Training / Testing Set 24), 2 (Training / Testing Set 25) and 3 (Training / Testing Set 26)

Model	Training / testing set	N^{H1}	RMSE (EC units)	AAPE (%)	AAE (EC units)
SMB_M_A_86_24_91	24	15	49.7	7.3	37.9
SMB_M_A_87_24_91	24	20	45.2	6.2	35.4
SMB_M_A_88_24_91	24	35	44.0	5.7	32.0
SMB_M_A_89_25_91	25	5	44.6	5.7	32.0
SMB_M_A_90_25_91	25	15	47.5	5.7	34.1
SMB_M_A_91_25_91	25	30	49.5	6.3	35.1
SMB_M_A_92_26_91	26	15	47.9	5.9	39.0
SMB_M_A_93_26_91	26	20	46.5	5.8	33.6
SMB_M_A_93_26_91	26	35	52.2	7.3	40.0

3.10.5 Results / Discussion

The RMSEs of the best real time forecasts for the models using the different training / testing sets (models SMB_M_A_88_24_91, SMB_M_A_89_25_91 and SMB_M_A_93_26_91), and the corresponding time taken for training (i.e. the time taken to process 100,000 training samples), are shown in Figure 3.175. It can be seen that the generalisation ability of the three models is very similar. The RMSE of the model with inputs obtained using method 1 was 44.0 EC units, compared with RMSEs of 44.6 EC units and 46.5 EC units when methods 2 and 3 were used. However, there is a marked difference in the time taken for training. It should be noted that there is good agreement between training time and the number of network inputs (and hence the

network size that gives optimum generalisation ability). In other words, the network that used the largest number of inputs (i.e. the network using the inputs obtained with the aid of method 1) took longest to train, whereas the training time for the network with the smallest number of inputs (i.e. the network using the inputs obtained with the aid of method 2) was shortest.

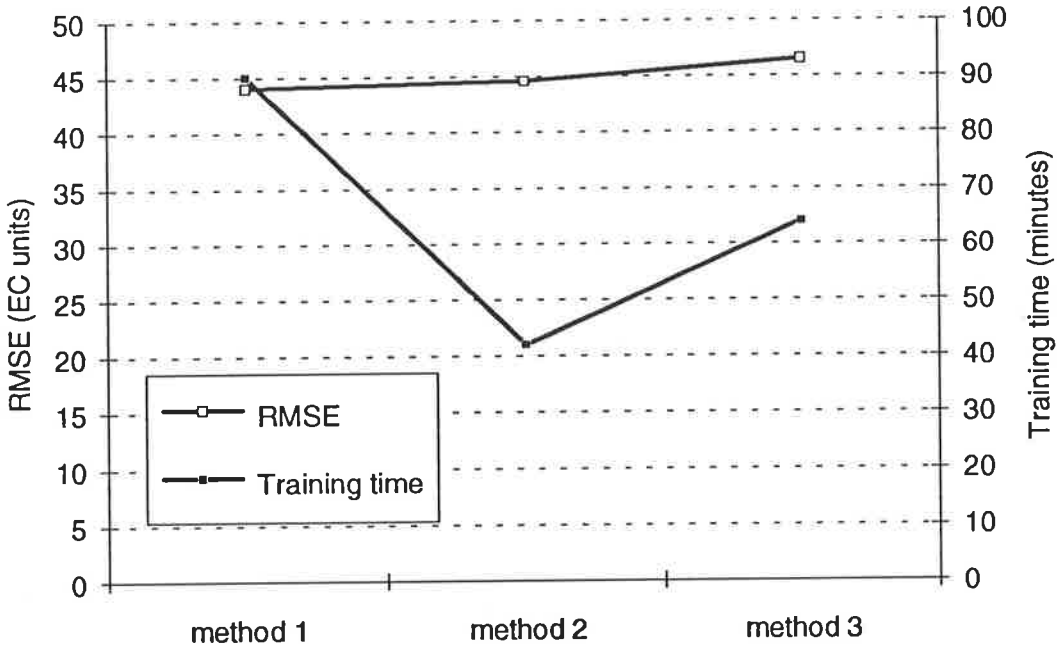


Figure 3.175: Best Real Time Forecasts and Corresponding Training Times For Models With Inputs Obtained Using Methods 1, 2 and 3

The results obtained in this section indicate that the method of Haugh and Box (method 1) and the method based on a neural network approach (method 2) provide suitable procedures for determining the inputs to multivariate ANN models. The results of the real time forecasting simulation indicate that the generalisation ability of the models using inputs obtained with the aid of methods 1 and 2 is almost identical, and compares favourably with that of the models using inputs obtained by using *a priori* knowledge about the physical processes resulting in high levels of salinity.

Although the generalisation ability of the models trained with inputs obtained using methods 1 and 2 was almost identical, there are a number of advantages associated with using the neural network based approach (method 2), including:

1. It can determine which inputs are significant for a specific forecasting period (e.g. 14 days in the case study considered), thus reducing the number of inputs, which in turn reduces network size and training time.

2. It can provide valuable information about the relationship between the input and output time series (e.g. salinity travel times between Murray Bridge and various upstream locations under a variety of flow conditions in the case study considered).
3. It is simpler and quicker to use, as there is no need for pre-processing of the data (e.g. differencing and pre-whitening).

A disadvantage of method 2 is that there is some judgement involved in determining the initial lags of the inputs (i.e. k_{max}), the network geometry and parameters as well as the level of significance above which inputs of the univariate and bivariate models are included in the multivariate model.

3.11 Conclusions

The results obtained in this chapter indicate that the generalisation ability of ANN models is relatively insensitive to the addition of less significant model inputs. This is because ANNs are data driven and have the ability to determine which of the inputs are significant. However, if the number of model inputs, and hence the size of the network, is too large, the results obtained might be adversely affected, possibly due to the difficulty in finding a good local minimum in the error surface. In addition, larger networks require more training data to efficiently optimise the connection weights and take longer to train.

Consequently, it is desirable to have a methodology to determine the order of the inputs (i.e. which lagged values to include from each input time series) for multivariate ANN models, so that the number of unnecessary model inputs can be reduced. This is especially true when no *a priori* knowledge is available to suggest possible inputs, and for complex problems, where the number of potential inputs is large.

The method of Haugh and Box (1977), which is commonly used to determine the inputs for multivariate time series models of the ARMA type, was found to be an appropriate method for determining the inputs for multivariate ANN models. In addition, a new method for determining the inputs for multivariate ANN models, which is based on a neural network approach, was introduced. The method involves the development of simple univariate and bivariate ANN models to establish relationships between the output time series and each of the input time series. The strength of these relationships can then be determined with the aid of a sensitivity analysis, and the inputs that have a significant effect on the outputs chosen as inputs to the multivariate model.

The results of a real time forecasting simulation, in which 14 day forecasts were obtained for 1991, indicate that the generalisation ability of the models using inputs obtained with the aid of the method of Haugh and Box and the neural network based approach was almost identical, and compares favourably with that of models using inputs that were obtained by using *a priori* knowledge about the physical processes resulting in high levels of salinity.

Despite the fact that the generalisation ability of the models trained with inputs obtained using the method of Haugh and Box and the neural network based approach was almost identical, using the latter method has a number of advantages, including:

1. It can determine which inputs are significant for a specific forecasting period (e.g. 14 days in the case study considered), thus reducing the number of inputs, which in turn decreases network size and training time.
2. It can provide valuable information about the relationship between the input and output time series (e.g. salinity travel times between Murray Bridge and various upstream locations under a variety of flow conditions in the case study considered).
3. It is simpler and quicker to use, as pre-processing of the data (e.g. differencing, pre-whitening) is not required.

The major disadvantage of the neural network based approach is that there is some judgement involved in determining the level of significance, above which inputs of the univariate and bivariate models are included in the multivariate model.

The results obtained in this chapter indicate that the choice of the training and testing data (i.e. which year(s) of data is/are used for training and which year(s) is/are used for testing) has a significant effect on the generalisation ability of multivariate ANN models. Consequently, it is vital to choose the testing set carefully. The data used for testing should be representative of the general relationship the model is trying to approximate. Training several models, and using different data for training and for testing in each, was found to be a good way of determining the generalisation ability of a model, as it minimises the significance of choosing particular training and testing sets, while maximising usage of the available data.

For the majority of ANN models trained, generalisation ability increased steadily with increased training, until a local minimum in the error surface was reached. Continued training resulted in oscillations in the RMS forecasting error with each weight update, the magnitude of which was a function of the size of the steps taken in weight space.

The transfer and error functions had a significant effect on the generalisation ability of the networks. The hyperbolic tangent transfer function and the quadratic error function were found to perform best. When a linear transfer function and the cubic and quartic error functions were used, inferior local minima in the error surface were reached. The learning rate, momentum and epoch size did not have a significant effect on generalisation ability.

Training speed was greatly affected by the size of the steps taken in weight space, which is a function of epoch size (depending on the learning rule used), learning rate, momentum, the error function and the gain of the transfer function. When larger steps are taken in weight space, learning speed, as well as the ability of networks to escape local minima in the error surface, are increased. However, the magnitude of the oscillations in RMSE forecasting error, once a local minimum in the error surface has been reached, is also greater, which can cause problems in real time forecasting applications when limited data are available. In addition, if the step size is too large, divergent behaviour may occur, or the network may cease learning.

When using cross-validation as the stopping criterion for real time forecasting simulations, it is necessary to have separate training, testing and forecasting sets. The network that gives the best forecasts for the testing set should be used for forecasting. In order to obtain good generalisation ability, and hence good real time forecasts, the testing set has to be chosen carefully, so that it is representative of the relationship to be approximated.

When dealing with large networks, many local minima exist in the error surface, making it difficult to obtain results close to the global optimum. However, the following procedure was found to be useful when trying to optimise the forecasts obtained using the testing set:

1. Train the network using a small epoch size, a relatively large learning rate and a relatively large momentum. It should be noted that absolute values of learning rate and momentum chosen are dependent on the nature of the error surface. The learning rate and momentum chosen should also be small enough to avoid divergent behaviour.
2. Stop training and test network performance at regular intervals. It should be noted that smaller testing intervals increase the chances of obtaining networks with better generalisation ability. It is advisable to reduce the testing interval once a network approaches a local minimum in the error surface.
3. Continue training until a local minimum in the error surface has been reached.

4. Select the network that produced the best forecast and continue training after reducing the learning rate and momentum to very small values and setting the epoch size equal to the number of samples in the training set.
5. Stop training and test the performance of the network after each weight update (i.e. the testing interval should be equal to the epoch size).
6. Continue training until there is no further improvement in the forecasts obtained.

When limited data are available, it might be better to present a fixed number of training samples to the network during training, in order to utilise all the available data in the training phase. However, it is vital to carry out some exploratory analysis, in which a small portion of the training set is used to test the performance of the network at various stages of learning. This gives an indication of how many training samples need to be presented to the network until a local minimum in the error surface is reached and of the magnitude of the oscillations in the RMS forecasting error, once a local minimum in the error surface has been reached, for a particular combination of network parameters. The knowledge obtained as part of the exploratory phase (i.e. what network parameters are appropriate and how many training samples should be presented to the network) can then be used to re-train the network, using all available data for training. The knowledge of when a local minimum in the error surface is reached may also be used to reduce the size of the steps taken in weight space at that point, in order to reduce the magnitude of the oscillations in the RMS forecasting error.

Although several researchers (e.g. Baum and Haussler, 1989, Weigend et al., 1990) suggest that more training data are required to achieve good generalisation ability for larger networks, for the case study considered, generalisation ability was not affected by network geometry. A wide range of network geometries were tried, including networks with one and two hidden layers and different numbers of nodes per hidden layer. Training speed, on the other hand was affected by network geometry, as the time taken to update the connection weights is a function of the number of weights.

The presence of hidden layer nodes had a significant effect on the way data are processed. The results obtained indicate that the hidden layer nodes perform a task similar to differencing. When ANN models were trained using differenced data, the addition of hidden layer nodes had no effect on model performance, as all the underlying relationships in the data (e.g. trends, seasonal variations) had already been removed. However, when the models were trained with the raw data, the addition of hidden layer nodes affected the performance of the models, suggesting that the hidden layer nodes detect some of the underlying relationships present in the data. Consequently, when ANNs with hidden layers are used, there is no need to difference the data.

The results obtained indicate that when multi-step forecasts are required, it is better to train networks for the required forecasting period directly, instead of training them for a one-step forecast and then applying it recursively. When the multi-step forecasts are obtained directly, multiple output nodes can be used; one for each required forecasting period. In the case study considered, network performance was not significantly affected by the number of network outputs. However, one would expect that the presence of a large number of output nodes could make it difficult to find a good local minimum in the error surface, as a result of the increased number of connection weights.

A comparison of the RMSEs of the 1, 5 and 14 day real time forecasts obtained for 1991 using the univariate ANN (UANN), multivariate ANN (MANN), univariate time series (ARIMA), multivariate time series (VARIMA) and a naive model (i.e. $z_{t+g} = z_t$, where g = the forecasting period) is given in Figure 3.176. It can be seen that:

1. Both multivariate models performed better than their univariate counterparts, suggesting that there is an advantage in using causal variables. However, the performance of the VARIMA model was only marginally better than that of the ARIMA model, whereas the performance of the MANN model was significantly better than that of the UANN model. The reason for this is that in ARMA type models, the model coefficients are estimated by maximising the likelihood function for the one day forecast, for which causal variables are less critical.
2. The ARMA type models performed better than the naive model for all forecasting periods.
3. The ANN models performed worse than the naive model for a forecasting period of 1 day, suggesting that ANN models are not suited to short term forecasting. However, as the forecasting period increases, the ANN models performed better than the naive models and the ARMA type models, indicating that ANN models are better suited to longer term forecasting than ARMA type models. Possible reasons for this include that ANN models base their forecasts on the underlying relationships that generated the time series, and that they can be trained for a specific forecasting period directly, without having to rely on a recursive forecasting procedure.

Of the two types of models considered, the ANN models were far simpler to develop. When developing the ANN models, there was no need to difference the data, as ANNs are able to account for non-stationarities in the data with the aid of their hidden layer nodes. In addition, using the direct method for obtaining multi-step forecasts is simpler than using the recursive method. Other advantages ANN models have over ARMA type models include that they are less sensitive to the number of model inputs and have superior generalisation ability when limited data sets are available.

A potential advantage of ARMA type models is that a direct mathematical relationship between the input and output variables is obtained. However, by carrying out a sensitivity analysis, the magnitude and the strength of the relationship between the model inputs and the model outputs that has been determined by an ANN model can be easily obtained.

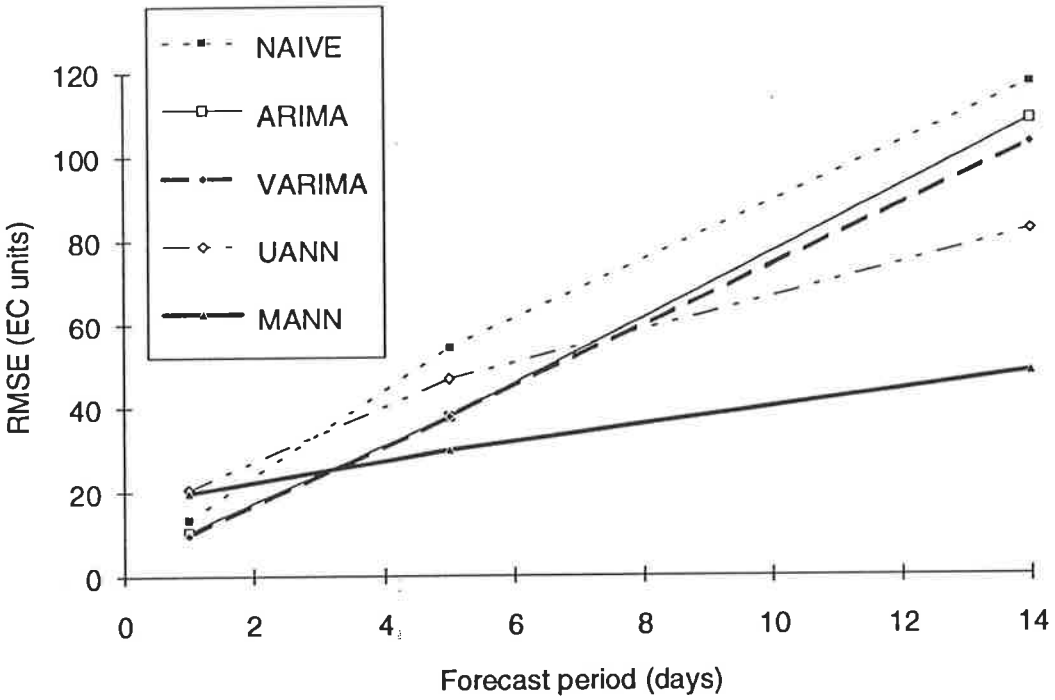


Figure 3.176: Comparison of the RMSEs of the Best Real Time Forecasts Obtained for 1991 Using the ARIMA, VARIMA, UANN, MANN and NAIVE Models

As can be seen from Figure 3.176, by far the best 14 day forecast of salinity in the River Murray at Murray Bridge was obtained using the multivariate ANN (MANN) model. The RMSE and the AAPE of the 14 day forecast were 48.1 EC units and 6.4% respectively. The 14 day forecast was very useful, as all major variations in salinity, including sharp peaks, were predicted without appreciable lag.

As multivariate ANN models are the model type most suitable for longer-term forecasting (Figure 3.176), this type of model was used to obtain the 28 day forecasts required to alter the schedules for pumping water from Murray Bridge to Adelaide taking salinity considerations into account. The 28 day forecasts obtained were very useful, as all major variations in salinity were predicted without appreciable lag, although the sharp peaks in salinity at times of high flow were not predicted. The RMSE and AAPE of the 28 day forecast were 78.3 EC units and 9.8% respectively.

Chapter 4

Modelling Blue-Green Algae in the River Murray

4.1 Introduction

Blue-green algae (cyanobacteria) are naturally occurring and form an integral part of healthy aquatic communities. Under certain conditions, population explosions, or blooms, of blue-green algae can occur. These blooms have a number of adverse effects on water users, as blue-green algae can produce toxins and reduce the aesthetic water quality.

Blue-green algae have become a major water quality issue in Australia, as a result of massive blooms in the Darling River, the River Murray and in the terminal lakes (Lakes Alexandrina and Albert) of the Murray. In South Australia, sites of particular concern include Murray Bridge, Mannum and Morgan, as water from these locations is pumped to most major South Australian cities, including Adelaide, Port Pirie, Port Augusta and Whyalla, for domestic and industrial consumption.

Treating water affected by blue-green algae is expensive, as advanced treatment processes are required to remove algal toxins. Consequently, it is vital to gain a better

understanding of the environmental conditions that trigger algal blooms, so that appropriate measures can be taken to prevent algal blooms from occurring. Computer modelling is one way of gaining a better understanding of these conditions.

In Chapter 3, multivariate ANN models were found to be a useful tool for obtaining forecasts of salinity in the River Murray at Murray Bridge up to 28 days in advance. In this chapter, multivariate ANN models are used to model the incidence of a species group of the cyanobacterium *Anabaena spp.* in the River Murray at Morgan. This is carried out in order to assess the suitability of ANN models for different types of water quality modelling problems. The purpose of the salinity forecasting case study was to obtain long term salinity forecasts. The processes which affect salinity levels at a particular location are well understood, and longer-term salinity forecasts can be obtained by considering river flows and past salinities at the location of interest and at several upstream sites. In contrast, the primary purpose of the case study considered in this chapter is not to obtain longer-term forecasts, but to determine a relationship between the incidence of *Anabaena* and some of the prevailing environmental conditions. Currently, this relationship is not well understood. Determining whether multivariate ANN models are capable of forecasting concentrations of *Anabaena* several weeks in advance, to give prior warning of impending blooms, is a secondary objective of the case study.

The information obtained in Chapter 3, regarding the development of ANN models for modelling multivariate time series, was utilised in this case study. However, no new investigations into the best procedures for developing multivariate ANN models were carried out. Consequently, raw data and the direct method of forecasting were used. The neural network based approach was used to determine appropriate model inputs and the effect of different internal parameters and network geometries was not investigated.

As the primary purpose of this case study is to investigate whether multivariate ANN models are capable of successfully predicting the incidence of *Anabaena* at Morgan, given the prevailing environmental conditions, there was no need to carry out a real time forecasting simulation.

4.2 Background

Algae are naturally occurring organisms of the kingdom Protista that form an integral part of healthy aquatic communities. With regards to water quality, freshwater algae

can be divided into four groups (Mackay and Eastburn, 1990), including diatoms, blue-green algae (cyanobacteria), flagellates and green algae (Figure 4.1).

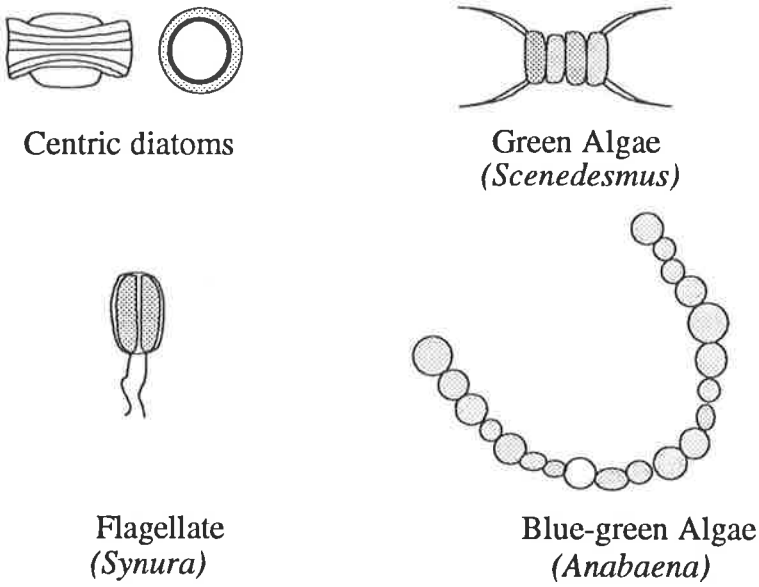


Figure 4.1: Examples of the Four Main Groups of Algae
(Source: Adapted from Mackay and Eastburn, 1990)

The distribution of algae in streams is dominated by the flow regime. There are distinct trends as one moves down a stream. In upland streams, flow is generally high, sediments are coarse, nutrient concentrations are low, the water is shallow and well mixed and light penetrates to the bottom. As the gradient flattens, flow decreases, sediments become finer and more organic, nutrient concentrations and water depth increase, vertical mixing decreases and light does not penetrate to the bottom.

At high flows, small, robust diatoms tend to dominate. Diatoms are dependent on high, turbulent flows to keep them in suspension and near the water surface. At intermediate flows, diatoms tend to settle to the bottom and green algae take over. At low flows, flagellates and cyanobacteria tend to dominate, as they are able to maintain their position in the water column; flagellates with the aid of their flagella and cyanobacteria with the aid of air vacuoles, which enable them to float at various depths in the water column.

In this research, the focus will be on cyanobacteria. Cyanobacteria were previously referred to as blue-green algae because of their size, form and life habit as photosynthetic bacteria. The main species of cyanobacteria which are most frequently associated with water quality problems in Australia include *Microcystis*, *Anabaena*, *Nodularia*, *Cylindrospermopsis* and *Oscillatoria*.

Under certain conditions, population explosions (blooms) of cyanobacteria occur, which can pose serious problems for water users. Such blooms are not a new phenomenon. In 1830, explorer Charles Sturt noted that the water in the Darling River had a taste of vegetable decay as well as a slight tinge of green (Creagh, 1992), which was the result of a bloom of blue-green algae. At present, there is some debate whether the frequency and intensity of blue-green algal blooms have increased in recent years.

4.2.1 Effects of Cyanobacterial Blooms

Blooms of cyanobacteria present problems for domestic, industrial, agricultural and recreational users of water (Burch, 1993). They produce toxins as well as undesirable tastes and odours.

4.2.1.1 Public Health Effects

Blooms of cyanobacteria can produce a variety of toxins including hepatotoxins, neurotoxins and endotoxins. The fact that cyanobacterial blooms can be toxic means that they present a public health risk and are not just a "nuisance" problem. A summary of the various toxins, their sources and effects is given in Table 4.1.

Algal toxins are primarily contained within the cells of the algae. They are only released to the surroundings in large quantities when the cell wall is ruptured, which may occur as part of the natural death of the algae or as a result of chemical treatment, such as the addition of copper sulphate (Kenefick et al., 1993; Jones and Orr, 1994; Bowmer et al., 1992).

The toxicity of blooms of cyanobacteria is highly variable, even if the same environmental conditions prevail (Creagh, 1992). A survey of Finnish fresh waters showed that toxins were present in 45% of 215 water bloom samples tested and that hepatotoxic blooms were twice as common as neurotoxic blooms (Sivonen et al., 1990). Even within a particular bloom, large spatial and temporal variations of toxin levels exist (Hrudey et al., 1994a). Extensive studies were carried out in Finland (Kiviranta et al., 1992; Luukkainen et al., 1993 and 1994; Namikoshi et al., 1990, 1992a, 1992b and 1992c; Sivonen et al., 1992a, 1992b, 1992c and 1992d) to determine the toxin composition of various blooms of *Anabaena*, *Microcystis* and *Oscillatoria*. It was found that the quantitative and qualitative variations of toxins were greatest among *Anabaena* and smallest among *Oscillatoria*. This high variability of toxin levels within blooms makes it difficult to accurately determine the average level of toxins in a bloom.

Table 4.1: Summary of Various Cyanobacterial Toxins and Effects (Source: Adapted from Burch, 1993)

Toxin	Toxin type	Source organism	Effects
Microcystin	Hepatotoxin	*Microcystis* *Anabaena*	• Hepatoenteritis • Liver damage • Tumor growth promotion
Nodularin	Hepatotoxin	*Nodularia*	• Hepatoenteritis • Liver damage • Tumor growth promotion
Cylindrospermopsin	Hepatotoxin	*Cylindro-spermopsis*	• Gastroenteritis • Hepatitis • Renal malfunctioning • Haemorrhaging
Anatoxin-a Anatoxin-a (s) Saxitoxin Neosaxitoxin	Neurotoxin	*Anabaena circinalis*	• Muscle tremors • Staggering • Paralysis • Breathing difficulties
Lipopolysaccharides	Endotoxin	Most cyanobacteria	• Gastroenteritis • Skin irritation • Eye irritation • Allergic reactions

Studies of the effects of various environmental factors on toxin production in blooms of *Nodularia*, *Oscillatoria* and *Anabaena* (Lehtimäki et al., 1994; Rapala et al., 1993; Sivonen, 1990) have shown that:

- High temperatures (30° C) generally decrease toxin production.
- Toxin production is at its maximum at light intensities most conducive to the growth of a particular species.
- Low phosphorus concentrations decrease hepatotoxin production.
- Neurotoxin production is independent of phosphorus concentrations.
- Low nitrogen concentrations favour toxin production in nitrogen fixing species.
- High nitrogen concentrations produce more neurotoxins and hepatotoxins in non-nitrogen fixing species.

Exposure of humans to cyanobacterial toxins may occur via the following mechanisms (Kuiper-Goodman et al., 1994; Hrudey et al., 1994b):

- Consumption of drinking water and algal health food tablets (oral route).

- Recreational use of rivers and lakes (oral or dermal route).
- Use of showers (inhalation).

A. Hepatotoxins

The hepatotoxins produced by blue-green algae include microcystin, nodularin and cylindrospermopsin (Burch, 1993) and are summarised in Table 4.1.

A1. Microcystin

Microcystin can be produced by *Microcystis* and *Anabaena*. At present, over 40 different microcystins have been identified (Drikas, 1994). These toxins include some of the most acutely lethal substances currently known (Hrudey et al., 1994b). Animal experiments have shown that lethal doses of microcystin-LR have resulted in structural damage to liver cells, causing massive haemorrhaging of the liver (Hooser et al., 1990). Sublethal levels of the toxin resulted in elevated levels of various liver enzymes in the blood.

The level of microcystins present in drinking water is usually quite low. At low exposure levels, experiments on animals have shown that microcystins:

- Affect the gastrointestinal tract (Falconer et al., 1992; Gayley et al., 1987; Jackson et al., 1984).
- Affect the kidney (Hooser et al., 1990; Radbergh et al., 1991).
- Affect the lungs (Hooser et al., 1989; Falconer et al., 1988; Falconer et al., 1981; Slatkin et al., 1983).
- May be very potent tumour promoters, stimulating the growth of potential liver and gut cancers (Falconer, 1991; Nishiwaki-Matsushima et al., 1992).
- Are very potent inhibitors of the phosphatase enzymes, which are vital for cell metabolism and control (Honkanen et al., 1990; MacKintosh et al., 1990).
- May be clastogenic, as chromosomal breakage was found to be increased when microcystins were present and was also dose related (Repavich et al., 1990).

Despite the high toxicity of microcystins, the lethal poisoning of humans due to microcystins in drinking water is not very likely (Hrudey et al., 1994b). In fact, there have been no recorded deaths of humans as a consequence of the consumption of drinking water poisoned by cyanobacteria in Australia (Murray Darling Basin Management Commission, 1993). As cyanobacterial blooms tend to occur repeatedly in the same water supply, it is more likely that microcystins have a chronic effect on the health of the people consuming the affected drinking water. This is particularly

significant, as microcystins have been found to be tumour promoters. As a guide, the maximum level of microcystins in a municipal water supply should not exceed 10 μg/L (Hrudey et al., 1994b).

A2. Nodularin

Nodularin is produced by *Nodularia* and has effects similar to those produced by microcystins. Nodularin causes hepatoenteritis, liver damage and tumour growth promotion (Burch, 1993).

A3. Cylindrospermopsin

Cylindrospermopsin is produced by *Cylindrospermopsis*. It is cytotoxic and the effects of oral consumption include the following (Falconer, 1994):

- Gastroenteritis as a result of injuries to the gut lining.
- Hepatitis as a result of injuries to liver cells.
- Renal malfunction as a result of injuries to kidney cells.
- Haemorrhaging as a result of injuries to blood vessels.

The clearance of cylindrospermopsin from the body is slower than that of microcystin or nodularin.

B. Neurotoxins

The neurotoxins produced by cyanobacteria include the Paralytic Shellfish Poisons (PSPs) anatoxin-a, anatoxin-a(s), saxitoxin and neosaxitoxin (Carmichael, 1994). In Australia, neurotoxins are produced by the various species of *Anabaena* (Table 4.1).

Rapid human death results if PSPs are consumed in sufficient quantities. In fact, the ingestion of contaminated shellfish has resulted in significant human morbidity and mortality (Luthy, 1979). PSPs act as neuromuscular blocking agents, resulting in muscle tremors, staggering, paralysis and breathing difficulties (Falconer, 1994). However, the clearance of neurotoxins from the body is quite rapid.

C. Endotoxins

Lipopolysaccharide (LPS) endotoxins are produced by most species of cyanobacteria (Burch, 1993). Endotoxins can result in gastroenteritis (Keleti et al., 1980) as well as

skin irritation, eye irritation, allergic reactions, hay-fever-like symptoms and asthma (Jones et al., 1993; Falconer, 1994).

D. Case studies

Documented cases of the health effects of cyanobacterial toxins have been made all around the world including Australia (Falconer et al., 1983a), Europe, Africa, Asia and North America (Codd et al., 1989; National Rivers Authority, UK, 1990). Examples of specific cases where cyanobacterial toxins have resulted in human illness are given below:

1. A major outbreak of gastrointestinal illness occurred in Sewickley, Pa, USA in August 1975 (Lippy and Erb, 1976; Sykora and Keleti, 1981). The disease affected 5,000 people out of a population of 8,000, and was caused by cyanobacterial toxins released by the treatment of a bloom of blue-green algae in the local water supply with copper sulphate.

2. Cases of human illness caused by drinking water contaminated with cyanobacterial toxins have also been reported in the prairies of Canada (Kuiper-Goodman, 1994). The toxins resulted in stomach cramps, vomiting, diarrhoea, fever, headache, pains in muscles and joints as well as general weakness (Dillenberg and Dehnel, 1959).

3. In Salisbury, East Africa, outbreaks of acute childhood gastroenteritis occurred around June and July each year from 1960 to 1965 (Zillberg, 1966). The illness could not be attributed to any infectious disease and the same problems did not occur in the neighbouring suburb, which had a different water supply. It was found that each year massive blooms of *Microcystis* occurred in the water supply of the affected suburb. These blooms died off naturally in mid-winter (around June and July) releasing the toxins.

4. Studies carried out by Yu et al. have found that in China, approximately 100,000 people die from primary liver cancer (PLC) each year (Harada, 1994). In several areas of the country, there was no correlation between the incidences of PLC and PLC-causing agents such as aflatoxin and the hepatitis B virus. In addition, the risk of contracting PLC was about 8 times higher for people who drank pond and ditch water than for people who drank well water (Yu, 1989). This suggests that the incidences of PLC might be related to the presence of microcystins in drinking water. A joint study between China and Japan was commenced in 1992 to

investigate the correlation between the incidences of PLC and the presence of microcystins in drinking water (Harada, 1994).

5. The worst case of poisoning of humans as a result of the presence of microcystins in drinking water in Australia occurred at Palm Island (a tropical island off the north-east coast of Australia) (Blyth, 1980). In November of 1979, over 100 aboriginal children got ill, presenting with symptoms of vomiting, headaches, abdominal pain, enlarged and tender livers, initial constipation and kidney damage. No deaths were recorded and all children recovered within a month. At the time of the outbreak of the disease, a massive bloom of cyanobacteria in the only water supply reservoir was killed after the addition of a large dose of copper sulphate. The predominant genus of blue-green algae present in the reservoir was *Cylindrospermopsis*, and tests using toxic water from the reservoir caused liver injury and tissue necrosis in mice (Hawkins et al., 1985).

6. Turner et al. (1990) report that in 1990, people canoeing on and swimming in an English reservoir affected by a bloom of *Microcystis* got sick, presenting with symptoms of vomiting, diarrhoea, sore throats, blistering around the mouth, abdominal pain and headaches.

4.2.1.2 Effects on Animal Health

The toxins discussed in Section 4.2.1.1 can have severe effects on animal health. Stock deaths resulting from cyanobacterial poisons have been widely reported in Australia and throughout the world.

The first recorded case of stock deaths occurred in Australia and was reported in 1878 (Francis, 1878). A bloom of *Nodularia Spumigera* occurred in Lake Alexandrina and resulted in the death of cattle, sheep, horses, pigs and dogs. Further stock deaths occurred in the vicinity of Lake Alexandrina in 1945, and 306 animals died as a result of a blue-green algal bloom in Lake Bonney in 1959 (State Water Laboratory, 1993). Other cases of stock deaths in Australia resulting from cyanobacterial toxins have been reported by Wood (1975) and May (1981).

4.2.1.3 Ecological Effects

The ecological effects of the toxins produced by blue-green algal blooms include:

- The poisoning of wild animals such as rodents, amphibians, fish, water fowl, bats, zebras, rhinoceros and zooplankton (Steffensen, 1991).
- The accumulation of toxins in shellfish (Luthy, 1979).

In addition, decaying algae increase the biological oxygen demand (BOD) of the water and consequently reduce the amount of oxygen available for other aquatic organisms, resulting in great stress and even death of these organisms, especially fish.

4.2.1.4 Aesthetic Effects

Blue-green algal blooms can have the following aesthetic effects on water bodies (MDBMC, 1993):

- They can cause undesirable tastes and odours.
- They can cause discolouration of the water.
- They can form unsightly, smelly scums on the water surface and along the shore.

Cyanobacteria are believed to produce metabolites, including geosmin and 2-methylisoborneol (MIB), which cause earthy/musty odours in drinking water (Table 4.2). Geosmin and MIB can be detected by humans at very low concentrations (10 to 30 ng/L), which is equivalent to less than 3 tablespoons of the compound in 200 Olympic swimming pools (Burch, 1993).

Table 4.2: Odour Causing Compounds and their Effects (Source: Adapted from Burch, 1993)

Compound	Source organism	Effects
Geosmin	*Anabaena* *Oscillatoria*	Earthy/musty odour
2-Methylisoborneol (MIB)	*Anabaena* *Oscillatoria*	Earthy/musty/camphorous odour

However, there is some controversy over whether the musty/earthy odours are produced by cyanobacteria or actinomycetes (Persson, 1992). The controversy is fuelled by the fact that the earthy/musty odours can occur without the presence of blue-green algal blooms and vice versa. Studies have shown that there is no correlation between geosmin and MIB and microcystin-LR (Hrudey et al., 1993; Kenefick et al., 1992). A possible explanation for this is that the geosmin and MIB, which are very stable compounds, are produced upstream by planktonic or benthic cyanobacteria and then transported considerable distances downstream by river flow (Hishida et al., 1988).

4.2.1.5 Effects on Water Supply Operations

The presence of large numbers of blue-green algae increases the suspended solids load in the water, which increases turbidity, reduces the efficiency of disinfection and can block filters and sprinklers (Steffensen, 1991). The presence of blooms of cyanobacteria can also result in significant diurnal fluctuations in pH, which can alter the form and toxicity of heavy metals and ammonia and the chemical reactions taking place as part of water treatment processes (Steffensen, 1991). In addition, blue-green algal blooms may require the provision of alternative water supplies (MDBMC, 1993).

The decay of blue-green algae increases the dissolved organic carbon load, as well as the biochemical oxygen demand (Steffensen, 1991). This increases:

- The amount of flocculants used during water treatment.
- The amount of chlorine required for disinfection.
- The potential for trihalomethane production.
- The potential for bacterial regrowth within the distribution system.

4.2.1.6 Effects on Recreation and Tourism

Blue-green algal blooms may prohibit the use of water bodies for recreational activities (e.g. swimming, windsurfing and water skiing). This may also affect tourism and may result in considerable economic losses to tour operators, caravan park owners and local shop keepers.

4.2.2 Addressing the Cyanobacteria Problem

There are two ways in which the cyanobacterial bloom problem can be addressed, namely by treating the affected water or by preventing the blooms from occurring in the first place.

4.2.2.1 Water Treatment

Conventional water treatment processes are capable of removing algal cells (at a cost), but are generally unable to remove algal toxins and odour-producing compounds. For example, the taste and odour problems associated with blooms of *Oscillatoria* in Lake Mjøsa (Norway) rendered the water undrinkable, even after the water had been treated (Baalsrud, 1982). Advanced treatment processes such as oxidation and the use of activated carbon filters are required to remove algal toxins and odour producing compounds. However, upgrading existing water treatment plants to account for the

removal of algal toxins and odour producing compounds is expected to run into hundreds of millions of dollars (MDBMC, 1993).

A. Removal of Algae

Algal cells may be removed using a number of techniques (MDBC, 1993), including:

- Physical separation methods (e.g. microstraining, microfiltration).
- Coagulation methods (e.g. coagulation/filtration/disinfection, SIROFLOC clarification).
- Flotation methods (e.g. dissolved air flotation, foam flotation /dispersed air flotation).
- Biological methods (e.g. grazing by zooplankton).

Guidelines based on the current state of knowledge (MDBC, 1993) recommend the following methods for algal cell removal:

- Dissolved air flotation plus filtration for towns.
- Dissolved air flotation or slow sand filtration for small communities.
- Slow sand filtration for livestock.
- Dissolved air flotation plus filtration for portable supplies.

B. Removal of Toxins

Although some conventional water treatment techniques are effective in removing algal cells, they are generally unable to remove the associated toxins. Algal toxins usually have to be removed using post-treatment methods such as oxidation or adsorption (Drikas, 1994).

B1. Oxidation

The use of various oxidants such as chlorine, ozone, chloramines and potassium permanganate has been assessed. The effectiveness of each of these is discussed in detail below and summarised in Table 4.3.

Chlorine:

Nicholson et al. (1993, 1994) have shown that aqueous chlorine and calcium hypochlorite were effective in rapidly destroying the algal toxins microcystin and nodularin. However, the effectiveness of the oxidants was found to be pH dependent and was greatly reduced above pH 8. In addition, a chlorine residual of at least 0.5 mg/L was required after a contact time of 30 minutes.

Table 4.3: Effectiveness of Various Oxidants in Toxin Removal

Oxidant	Toxins removed			Formation of noxious byproducts
	Hepato-toxins	Neuro-toxins	Endo-toxins	
Chlorine	Yes	No	?	?
Ozone	Yes	Yes	?	?
Chloramine	No	?	?	?
Potassium Permanganate	Yes	?	?	?

? = Not yet investigated

Several researchers (Hoffmann, 1976; Keijola et al., 1988; Himberg et al., 1989) found chlorine to be ineffective for the removal of algal toxins. However, in all cases the chlorine to toxin ratio used was small, so that insufficient chlorine was present to react with the toxins.

The conditions that resulted in the rapid destruction of the hepatotoxins microcystin and nodularin (i.e. pH less than 8 and a chlorine residual of 0.5 mg/L) were ineffective for the removal of the neurotoxin anatoxin-a (Rositano and Nicholson, 1994).

The formation of noxious byproducts as a result of the chlorination of the hepatotoxins has not yet been assessed and needs to be carried out in the future.

Ozone:

Ozone was found to be effective in removing both hepatotoxins and anatoxin-a within very short contact times (Keijola et al., 1988; Himberg et al., 1989; James and Fawell, 1991; Rositano and Nicholson, 1994). It was also found that the removal efficiency was dependent on ozone dose when the dose was below the ozone demand of the water. However, once the ozone demand of the water had been met, toxin removal was complete.

Ozone has also been found to be effective for the removal of nodularin (Pieronne, 1993). However, the effectiveness of ozone in removing endotoxins has not yet been assessed. In addition, the formation of noxious byproducts as a result of the ozonation of algal toxins needs to be assessed.

Others:
Nicholson et al. (1994) found that chloramines were ineffective in removing algal toxins, even after contact times of up to five days.

Potassium permanganate has been found to be effective in the removal of microcystin-LR (Rositano and Nicholson, 1994). However, more research needs to be carried out to assess its effectiveness in removing other algal toxins, its pH dependence and the formation of noxious byproducts.

B2. Adsorption

Activated carbon was found to be an effective means of removing algal toxins by a number of researchers (Hoffmann, 1976; Falconer et al., 1983b and 1989; Keijola et al., 1988; Himberg et al., 1989; James and Fawell, 1991; Donati et al., 1993).

However, the adsorption capacity depends on the type of activated carbon used. Drikas (1994) found that wood based carbons were clearly the most effective adsorbents, followed by coal based carbons, coconut based carbons and peat moss based carbons. A correlation was found between the mesopore volume of the activated carbon and its adsorptive capacity. Mesopore volume is primarily a function of the material used to produce the activated carbon: wood-based carbons have large mesopore volume, coconut-based carbons have low mesopore volume and coal-based carbons fall somewhere in between (Drikas, 1994). As algal toxins generally consist of large molecules, better adsorption occurs when using activated carbons with large mesopore volumes. A similar correlation between adsorption capacity and mesopore volume was found by Donati et al. (1994).

In natural waters, a reduced effectiveness of activated carbons in adsorbing algal toxins can be expected as natural organics compete with the algal toxins for adsorption.

B3. Others

Slow sand filtration has been shown to remove some cyanobacterial toxins (Himberg et al., 1989; Keijola et al., 1988).

C. Removal of Odour-Producing Compounds

The primary odour-producing compounds, geosmin and 2-methylisoborneol, are resistant to oxidation unless the most powerful oxidising agents are used (Wnorowski,

1992). Consequently, the most effective methods for the removal of the odour-producing compounds include advanced oxidation techniques (Koch et al., 1992) or a combination of these with the use of biological filters (Egashira et al., 1992).

4.2.2.2 Prevention of Algal Blooms

As discussed in Section 4.2.2.1, the removal of algal toxins and odours is expensive, and the side effects of some of the methods used for the above purposes have not yet been fully investigated. Consequently, the best way of addressing the blue-green algal problem is to prevent algal blooms from occurring. One way to achieve this is to determine the causal factors of algal blooms, and to subsequently put management strategies into place to control these factors.

The size of the algal population at any particular point in space and time is a function of the amount of algae imported, the amount of algae exported, the amount of algae lost through decomposition and grazing and the algal growth rate (Hodgkin et al., 1980). In the South Australian reaches of the River Murray, the amount of algae imported and exported during summer and autumn is generally low, thus restricting a specific algal population to a particular section of the River (Mackay and Eastburn, 1990). Consequently, algal blooms are generally caused by conditions conducive for large algal growth rates.

Factors which determine whether blooms of a particular species of algae occur include the seed source, the general climatic conditions and the characteristics of the water body (Steffensen, 1991). Potential seed sources include lagoons and backwaters (MDBMC, 1993), but algal spores may also be re-suspended from the sediments at times of turbulent flow (Mackay and Eastburn, 1990). The algal growth rate depends on a number of climatic conditions and characteristics of the water body, including temperature, flow, pH, dissolved oxygen, the availability of nutrients, the amount of light available and the state of the aquatic ecosystem (MDBMC, 1993).

The interaction of the above factors is very complex and the relative significance of each of the parameters is catchment dependent. However, the actual combination of factors that will trigger algal blooms is unknown, and identical conditions can have apparently contradictory effects in different locations (Creagh, 1992). In order to gain a better understanding of the mechanisms that trigger algal blooms, real-time instream water quality monitoring might be required (MDBC, 1993).

A. Light

Light is essential for photosynthesis and hence a requirement for algal growth. The availability of light is a function of the intensity and duration of the incoming solar radiation as well as the transparency of the water. The transparency of the water depends on its turbidity and colour. High turbidity and colour limit the amount of light that can penetrate the water column and may thus prevent the rapid growth of algae.

In the South Australian reaches of the River Murray, the variation of turbidity with time exhibits the following properties (Croome, 1980; Mackay and Eastburn, 1990):

- Turbidity varies considerably on a seasonal basis as well as from year to year.
- There is a strong positive correlation between fluctuations in flow and turbidity fluctuations.
- Flows from the Darling River have a disproportionate influence on downstream turbidities, as the water in the Darling is considerably more turbid than that in the upstream reaches of the Murray. When the flow in the Darling is greater than approximately 200 GL/month, the turbidity of the River Murray in South Australia is high, resulting in a reduction in the number and taxa of blue-green algae.

It should be noted that cyanobacteria have the ability to position themselves in the water column in the region of optimum light levels with the aid of their air vacuoles.

B. Flow

Low flow conditions are favourable for the growth of blue-green algae for the following reasons (MDBMC, 1993):

- They increase the retention time of the water in the river, thus increasing the time available for algal blooms to develop.
- They decrease the turbulence of the water, which helps the algae maintain their optimum position in the water column.
- They are associated with lower turbidity, therefore enabling more light to penetrate.
- They decrease the amount of mixing that takes place in the water, which can lead to anoxic conditions near the river bed, releasing nutrients normally tied up in the sediments.

Low flow conditions exist in the South Australian reaches of the River Murray during summer and autumn. The frequency and duration of the low flow periods have increased as a result of flow regulation and the amount of water extracted for irrigation

purposes. This is thought to be one of the factors responsible for the increase in the intensity and frequency of blooms of blue-green algae.

Although calm, low flow conditions are necessary for the growth of blue-green algae, turbulent conditions prior to the onset of the period of low flow may be beneficial to the growth of algae by transporting nutrients from upstream sites as well as re-suspending algal spores tied up in the sediment (Mackay and Eastburn, 1990).

C. Nutrients

Nitrogen and phosphorus are the nutrients which are most important for the growth of blue-green algae. Other nutrients that might play a significant role include carbon, silica, iron, sulphur and molybdenum (MDBC, 1993).

The major forms of phosphorus present in natural waters include dissolved inorganic phosphorus, dissolved organic phosphorus and particulate phosphorus (Croome, 1980). The major forms of nitrogen present in natural waters include nitrate, nitrite, ammonia and organic nitrogen (Glatz, 1994).

Nutrient enrichment of water bodies occurs naturally over long periods of time and is called eutrophication. However, human activity such as land clearing, agriculture, human settlement, the disposal of wastewater and industrial effluent has sped up the process of eutrophication (Creagh, 1992).

Sources of nitrogen and phosphorus are discussed by a number of authors (MDBC, 1993, Hodgkin et al., 1980; MDBMC, 1993). For the Murray-Darling basin, the major sources include:

1. **Domestic and industrial wastewaters**: Sewage effluent is the largest nutrient point source. At times of low flow, during which the risk of algal blooms is greatest, point sources are the major contributor of nutrients. The biggest input of phosphorus to sewage treatment plants (30% to 50%) comes from laundry detergents. The majority of the phosphorus present in sewage effluent is in soluble orthophosphate form and can therefore be readily used by algae.

2. **Irrigated agriculture**: Agricultural practices add to the nutrient loading in rivers via irrigation drains and runoff from agricultural land, both of which contain nitrogen and phosphorus from fertilisers.

3. **Intensive rural industries**: Intensive animal industries including piggeries, cattle feedlots and poultry farms produce wastes high in nutrient concentrations. However, these wastes are not allowed to be discharged into waterways. The wastes from processing industries such as canneries, dairy processing plants, tanneries, wool scouring plants and abattoirs, on the other hand, are being discharged into rivers. The wastes from fish farms are also rich in nutrients and are being discharged into rivers.

4. **Urban stormwater**: The main source of nutrients in urban stormwater comes from fertilisers. However, the phosphorus present in runoff is generally bound to clay particles and is thus not in a form that is easily usable by algae.

5. **Groundwater**: Nutrients are added to rivers by groundwater inflows. The nutrient concentrations in groundwater samples are generally high.

6. **River sediments**: During periods of high external nutrient loading, sediments are able to take up some of the nutrients and store them for long periods of time. The nutrients stored in the sediments can be released to the water column at a later date and is then available for use by algae. In the River Murray, sufficient nutrients are stored in the sediments to sustain algal blooms for years, even if the input of external nutrients were to stop immediately.

7. **Biomass**: The biomass constitutes an important nutrient pool.

In general, the majority of phosphorus entering surface waters comes from point sources, whereas the majority of nitrogen comes from diffuse sources (Croome, 1980). Phosphorus is usually the nutrient controlling the growth of blue-green algae, as some are able to fix nitrogen from the atmosphere. Consequently, blue-green algae have a distinct advantage over other algae in an environment where there is a plentiful supply of phosphorus, but a limited supply of nitrogen.

In the South Australian reaches of the River Murray, the levels of oxidised nitrogen are usually low. In contrast, the levels of phosphorus are generally substantially greater than those required for algal growth, creating conditions favourable for the growth of nitrogen fixing cyanobacteria. However, after extended periods of low flow, phosphorus levels may become limiting (Croome, 1980).

In the River Murray, there is a strong correlation between total phosphorus and turbidity, as well as soluble phosphorus and turbidity. In addition, the proportion of

soluble phosphorus is much higher for water from the Darling than it is for water from the Murray (Croome, 1980).

D. Temperature

Temperature is an important factor affecting the rate of blue-green algal growth. Higher temperatures increase the rate of biological and chemical reactions and hence the rate at which algae grow. Low temperatures do not necessarily result in the death of the algae, but significantly reduce the rate at which they grow (Mackay and Eastburn, 1990).

E. The state of the aquatic ecosystem

A healthy aquatic ecosystem contains planktonic grazers, which eat the algae and digest the nutrients they contain (MDBMC, 1993). Changes in environmental conditions can disturb the balance in the ecosystem, resulting in the removal of some components of the food chain and excessive, unchecked growth of others. Factors which can result in such changes in the ecosystem include the draining of wetlands, the removal of native vegetation, poor management of the riparian zone and the dredging and desnagging of streams (MDBMC, 1993).

In the River Murray, the introduction of non-native fish such as the European carp and *Gambusia* has also had detrimental effects on the ecosystem (MDBMC, 1993), including the disturbance of sediments on the river bed, a decline in the number of water plants and a decline in the number of smaller native algal predators.

4.2.3 Modelling the Incidence of Algal Populations

A review of the different techniques used to model the incidence of algal populations is given by French and Recknagel (1994) and Recknagel et al. (1995a). The modelling techniques used can be divided into four main groups, namely statistical, physically based (deterministic), rule based (heuristic) and ANN models.

Traditionally, the statistical methods used to model the incidence of algal populations have been based on regression analysis. They relate the annual peak concentration (or biomass) of the total phytoplankton population (or a particular species of algae) to limiting environmental variables such as total phosphorus, total nitrogen, temperature, turbidity and water depth. This type of model generally uses cross-sectional (and sometimes time series) data collected from a number of lakes.

The most basic models of the above type relate chlorophyll *a* to concentrations of total phosphorus (e.g. Dillon and Rigler, 1974; Jones and Bachmann, 1976; Vollenweider, 1976). Smith (1985) developed a regression model that relates summer mean blue-green algal biomass to the concentration of total phosphorus and the mean lake depth. Smith et al. (1987) used stepwise multiple regression analysis to predict the summer peak biomass of four species of cyanobacteria, including *Aphanizomenon flos-aquae*, *Anabaena flos-aquae*, *Oscillatoria agardhii* and *Microsystis aeruginosa*. The potential input variables considered include summer mean temperature, mean depth, mean total nitrogen concentration, mean total phosphorus concentration, mean secchi depth transparency and mean carbon dioxide concentration in the water column. The non-nitrogen fixing species (i.e. *Oscillatoria agardhii* and *Microsystis aeruginosa*) were found to be better correlated to the environmental variables considered than the nitrogen fixing species (i.e. *Aphanizomenon flos-aquae* and *Anabaena flos-aquae*).

Physically based (deterministic) models endeavour to model the physical processes that affect the size of the phytoplankton population. Examples of physically based models that relate phytoplankton biomass to external nutrient load, temperature, light, mixing and zooplankton grazing in lakes and reservoirs include the models of Jorgensen (1976) and Behndorf and Recknagel (1982). The latter model was extended to cater for internal phosphorus loading from the sediments (Recknagel et al., 1995b).

Rule-based models (e.g. Reynolds, 1984) use knowledge about the factors that affect the size of the population of species assemblages of phytoplankton to qualitatively predict the seasonal phytoplankton composition. Fuzzy models (e.g. Recknagel et al., 1994) can be used to quantify the predictions of rule-based approaches.

None of the above approaches have been successful in predicting the magnitude and timing of the incidence of particular species or species groups of algae (French and Recknagel, 1994). However, recent studies have shown that ANNs appear to be a promising tool for solving the above problem. French and Recknagel (1994) used back-propagation ANNs to predict the incidence of 7 types of algae, given 9 physical and limnological variables in the Saidenbach Reservoir (Germany). Three years of data were used for training and two years of data were used for testing. Recknagel et al. (1995a) developed ANN models relating various algal species to a number of causal variables for Lakes Kasumigaura (Japan), Biwa (Japan) and Tuusulanjaervi (Finland) and the River Darling. The number of years of data used for training were 8, 6, 10 and 10 respectively. Two years of data were used for testing in each case.

All of the case studies which applied ANN methods to the modelling of algae used only two years of data for validation. This makes it difficult to accurately assess the generalisation ability of the models, as the results obtained can be significantly affected by which years of data are used for testing. In the case study considered in this chapter, a number of models are developed, each using different years of data for training and for testing, in order to get a better indication of the ability of ANNs to predict algal concentrations.

Only one of the above case studies considers the prediction of algae in a river (the Darling River), rather than a lake. The model inputs used include the latest known values (i.e. at time t-1) of a number of variables including information on available nutrients, flow and the transparency, temperature and conductivity of the water. However, as the river environment is a dynamic one (it is dominated by the flow regime), it is worthwhile to consider the inclusion of lagged inputs (i.e. at times t-1, t-2, ..., $t\text{-}k_{max}$). In this chapter, the effect of using lagged inputs on the ability of ANNs to forecast concentrations of blue-green algae in rivers is assessed.

Although a sensitivity analysis was carried out on the model predicting algal concentrations in the Darling River (Recknagel et al., 1995a), which provides useful information about the relative significance of each of the input variables, the sign of the sensitivities was not taken into account. Consequently, no information was obtained on which variables favour, and which inhibit, algal growth. In the research presented in this chapter, the sign of the sensitivities of each of the inputs is considered.

Another difference between the case study considered in this research and the one considered by Recknagel et al. (1995a) is that in this research, multi-step forecasts are obtained several weeks in advance, which gives prior warning of impending blooms to water management authorities.

4.3 Case Study

In the case study considered, multivariate ANN methods are used to model the incidence of a species group of the cyanobacterium *Anabaena spp.* in the River Murray at Morgan, South Australia (Figure 3.4). *Anabaena* were also predicted in the Darling River by Recknagel et al. (1995a). However, the results obtained were not presented. As discussed in Section 4.1, the primary objective of this case study is to obtain a relationship between the incidence of *Anabaena* and the prevailing environmental conditions, and the secondary objective is to forecast *Anabaena* concentrations several weeks in advance. The effect of using lagged inputs is also investigated.

Anabaena are the most prolific species of blue-green algae in the lower reaches of the Murray and are thus of most concern to water authorities. It is important to control the size of the blue-green algal population at Morgan, as a water treatment plant is located there and water abstracted at this point is delivered to the cities of Port Pirie, Port Augusta and Whyalla via the Morgan-Whyalla pipeline (Figure 4.2).

4.3.1 Available Data

A summary of the available data is given in Table 4.4. All data were available from 1983 to 1993 and were collected at Morgan; except for flow, which was measured just downstream of Lock 7 (Figure 3.4). All data were converted to weekly values using interpolation.

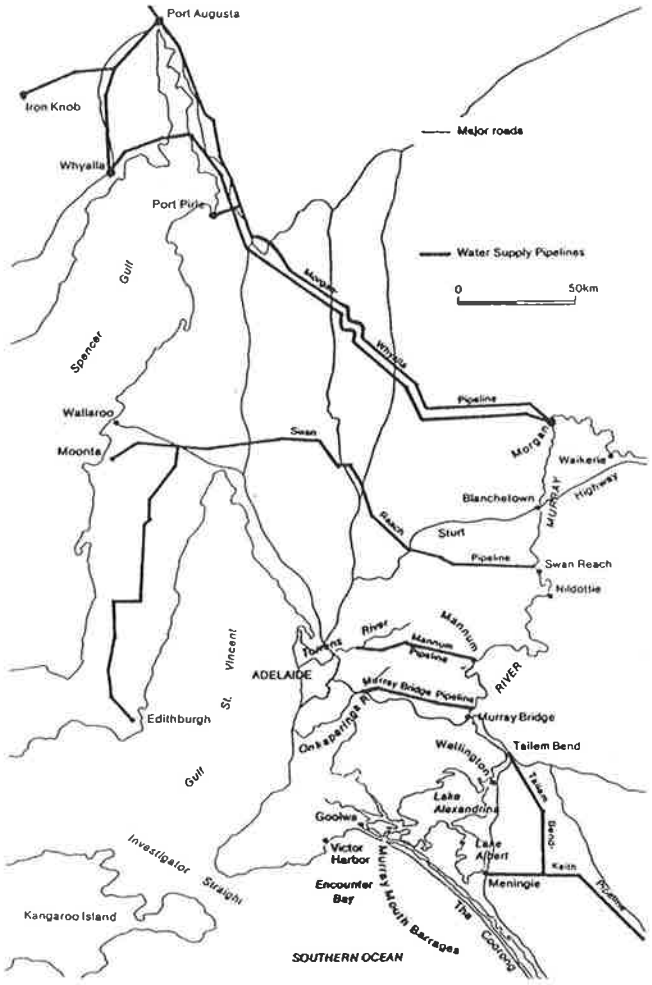


Figure 4.2: Location of Port Pirie, Port Augusta and Whyalla in Relation to Morgan

Table 4.4: Available Data

Variable	Units	Sampling interval
Anabaena spp.	cells/mL	weekly
Colour	HU	weekly
Turbidity	NTU	weekly
Temperature	°C	weekly
Flow	ML/day	daily
Soluble Phosphorus	mg/L	monthly
Total Phosphorus	mg/L	monthly
Nitrogen (Nitrite + Nitrate)	mg/L	monthly

The means, standard deviations, maxima and minima of the available time series from 1983 to 1993 are shown in Table 4.5

Table 4.5: Statistics of the Available Time Series (1983 to 1993)

Variable	Average	Standard Deviation	Maximum	Minimum
Anabaena spp. (cells/mL)	201	1,166	25,252	0
Colour (HU)	38.8	25.6	133	9.0
Turbidity (NTU)	82.0	60.3	410	15.0
Temperature (°C)	18.7	4.9	30.0	10.0
Flow (ML/day)	23,564	24,866	114,563	393
Soluble Phosphorus (mg/L)	0.03	0.03	0.19	0.0
Total Phosphorus (mg/L)	0.15	0.07	0.9	0.05
Nitrogen (mg/L)	0.11	0.09	0.53	0.01

It should be noted that no data were available on a number of factors which can influence the size of algal populations, including the amount of incoming radiation, the concentrations of algal predators and competing algal species, the thermal structure of the river, pH and dissolved oxygen. In addition, there are a number of sampling errors associated with the collection of the blue-green algal data (which is carried out by taking "grab samples" from the edge of the river), as the spatial distribution of the algae is expected to be non-uniform. The spatial distribution of blue-green algae is affected by wind speed and direction and the cyclic rising and falling of algal cells due to the collapse and regeneration of gas vesicles.

4.3.1.1 Inspection of Plots of Data

***Anabaena*:**

A plot of concentrations (on a log scale) of *Anabaena* at Morgan from 1983/84 to 1992/93 is shown in Figure 4.3. It should be noted that time will be expressed in Julian years henceforth, as blooms of *Anabaena* tend to occur at the end of the year. It can be seen from Figure 4.3 that significant events (concentrations ≥ 1000 cells/mL) occurred in 1985/86, 1987/88, 1989/90, 1990/91, 1991/92 and 1992/93. An arbitrary concentration of 1000 cells/mL was chosen as the level to distinguish between significant and non-significant events, as concentrations below 1000 cells/mL are regarded with less concern by water authorities (Burch, 1995). However, at concentrations in excess of 1000 cells/mL, the aesthetic quality of the water becomes unsatisfactory as a result of discolouration and the presence of unpleasant odours. In addition, the potential for the occurrence of algal blooms is increased greatly and water quality monitoring is increased.

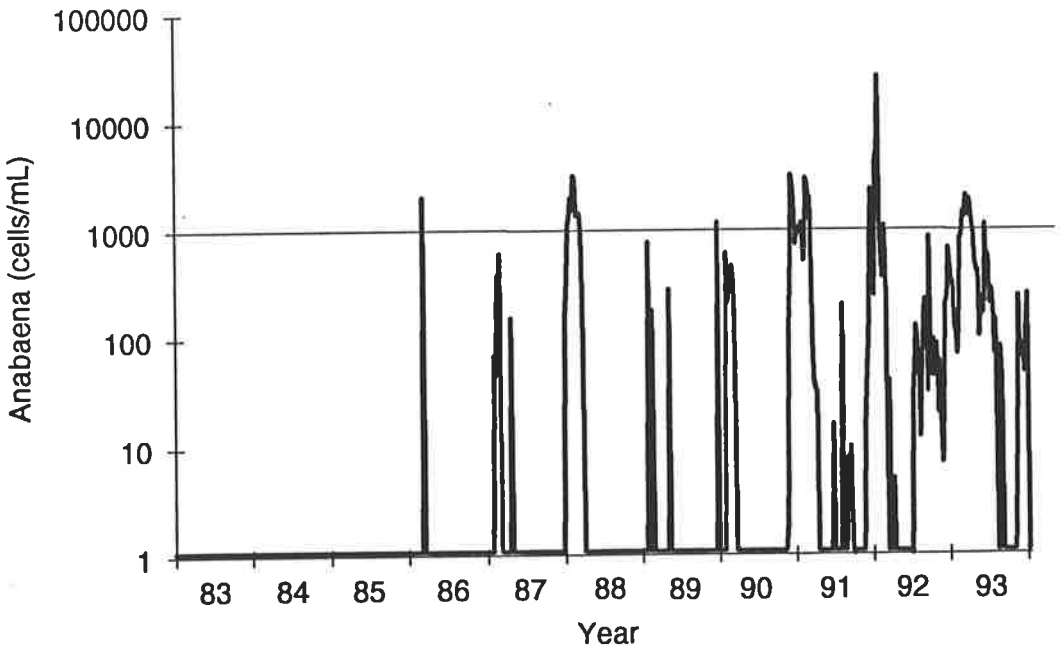


Figure 4.3: Concentrations of *Anabaena* at Morgan (1983 to 1993)

Figure 4.3 indicates the absence of *Anabaena* in 1983/84 and 1984/85. It is interesting to note that during these years (especially in 1983/84), the turbidity levels (Figure 4.5) were significantly higher than those in the other years. The largest bloom of *Anabaena* occurred in 1991/92, with concentrations of *Anabaena* far exceeding those in any other year.

Colour:

A plot showing the variation of *Anabaena* and colour at Morgan is shown in Figure 4.4. It can be seen that a negative correlation exists between *Anabaena* concentration and colour. This can be expected, as low levels of colour increase the amount of light that can penetrate, thus increasing the algal growth rate.

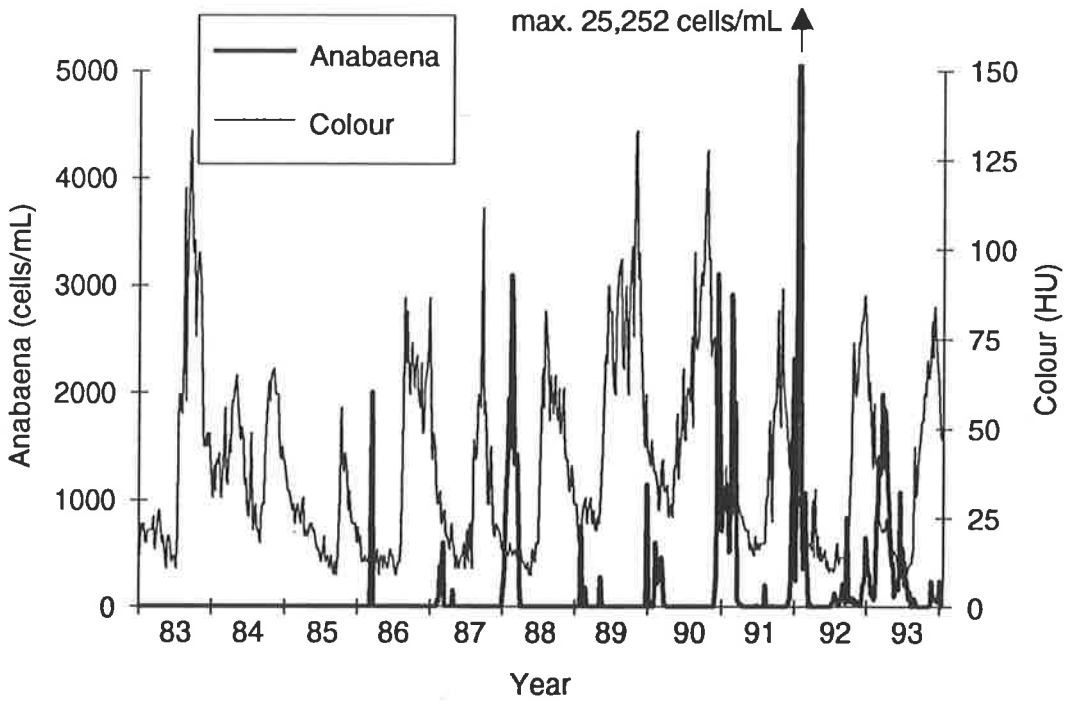


Figure 4.4: Concentrations of *Anabaena* and Colour at Morgan (1983 to 1993)

Turbidity:

A plot showing the variation of *Anabaena* and turbidity at Morgan is shown in Figure 4.5. It can be seen that a negative correlation exists between *Anabaena* concentration and turbidity. This can be expected, as low levels of turbidity increase the transparency of the water, thus allowing more light to penetrate and increasing algal growth rate. It is interesting to note that the turbidity levels in 1983/84 and 1984/85 were very high. This is because, during those years, the majority of the water in the South Australian reaches of the River Murray originated from the Darling River, which is considerably more turbid than the upstream reaches of the River Murray.

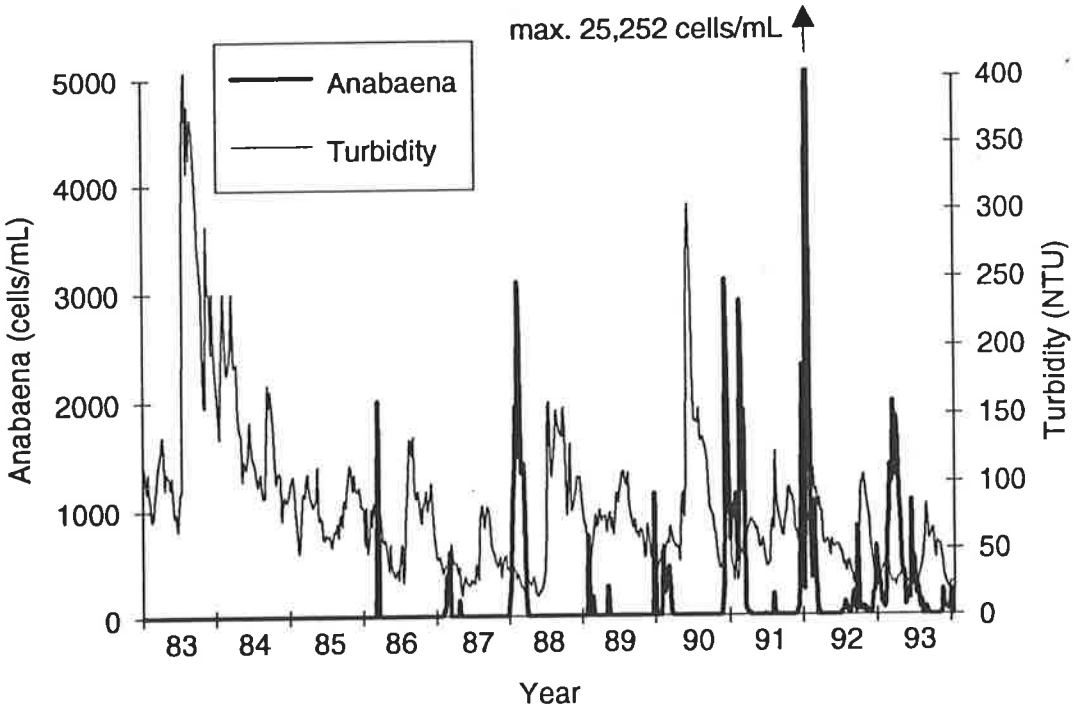


Figure 4.5: Concentrations of *Anabaena* and Turbidity at Morgan (1983 to 1993)

Temperature:

A plot showing the variation of *Anabaena* and temperature at Morgan is shown in Figure 4.6. It can be seen that a positive correlation exists between *Anabaena* concentration and temperature. This is to be expected, as the algal growth rate is increased at higher temperatures.

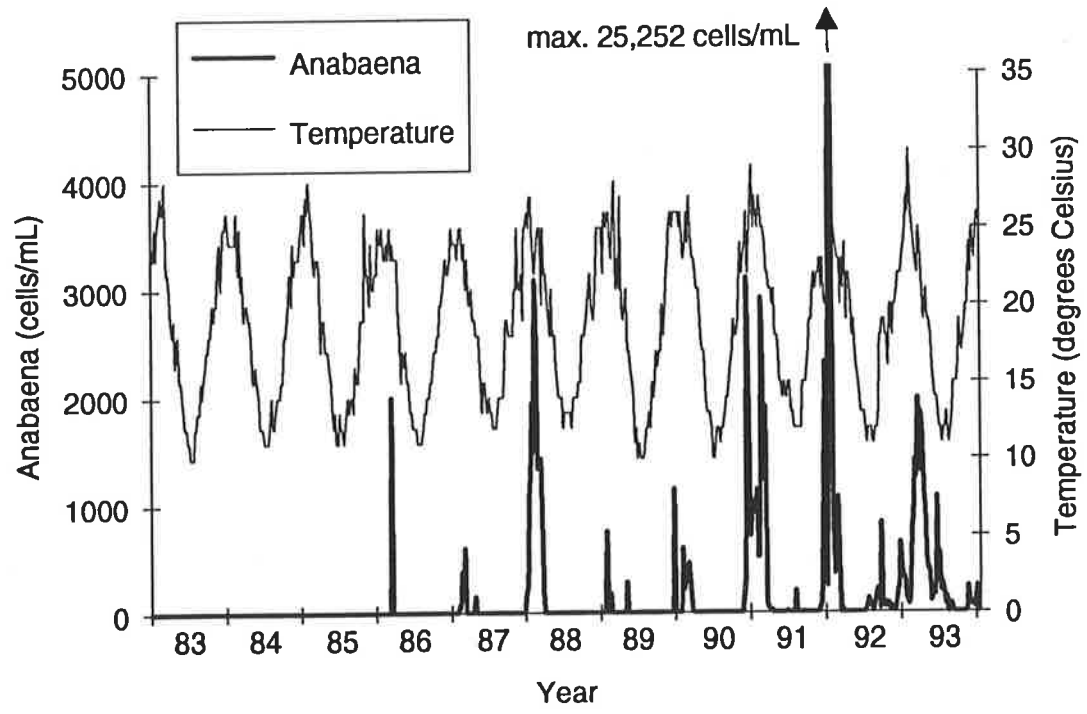


Figure 4.6: Concentrations of *Anabaena* and Temperature at Morgan (1983 to 1993)

Flow:

A plot showing the variation of *Anabaena* at Morgan and flow at Lock 7 is shown in Figure 4.7. It can be seen that a negative correlation exists between *Anabaena* concentration and flow. This can be expected, as low flow conditions favour the growth of blue-green algae, as discussed in Section 4.2.2.2.

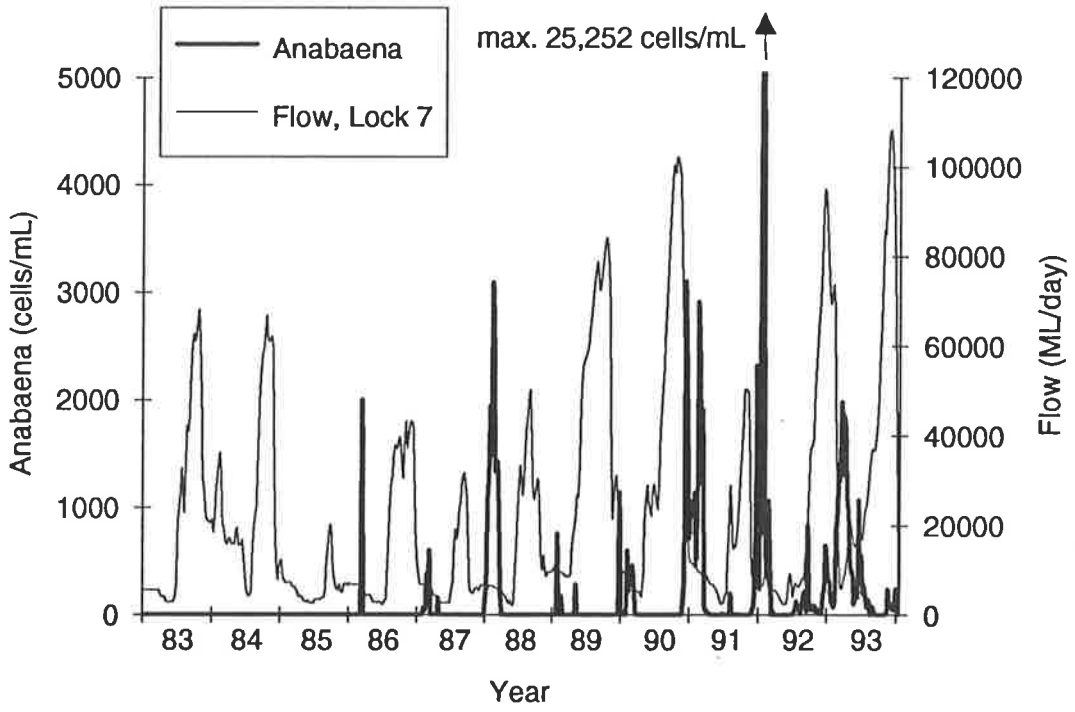


Figure 4.7: Concentrations of *Anabaena* at Morgan and Flow at Lock 7 (1983 to 1993)

Total phosphorus:

A plot showing the variation of *Anabaena* and total phosphorus at Morgan is shown in Figure 4.8. It can be seen that a negative correlation exists between *Anabaena* concentration and total phosphorus. This is contrary to what one would expect, as higher levels of nutrients are needed for the growth of algae. However, levels of phosphorus are strongly related to turbidity (Figure 4.9), as phosphorus is generally bound to clay particles responsible for high levels of turbidity. Consequently, one can conclude that the effects due to high turbidity are more significant than those due to the reduced availability of phosphorus, suggesting that phosphorus is not limiting.

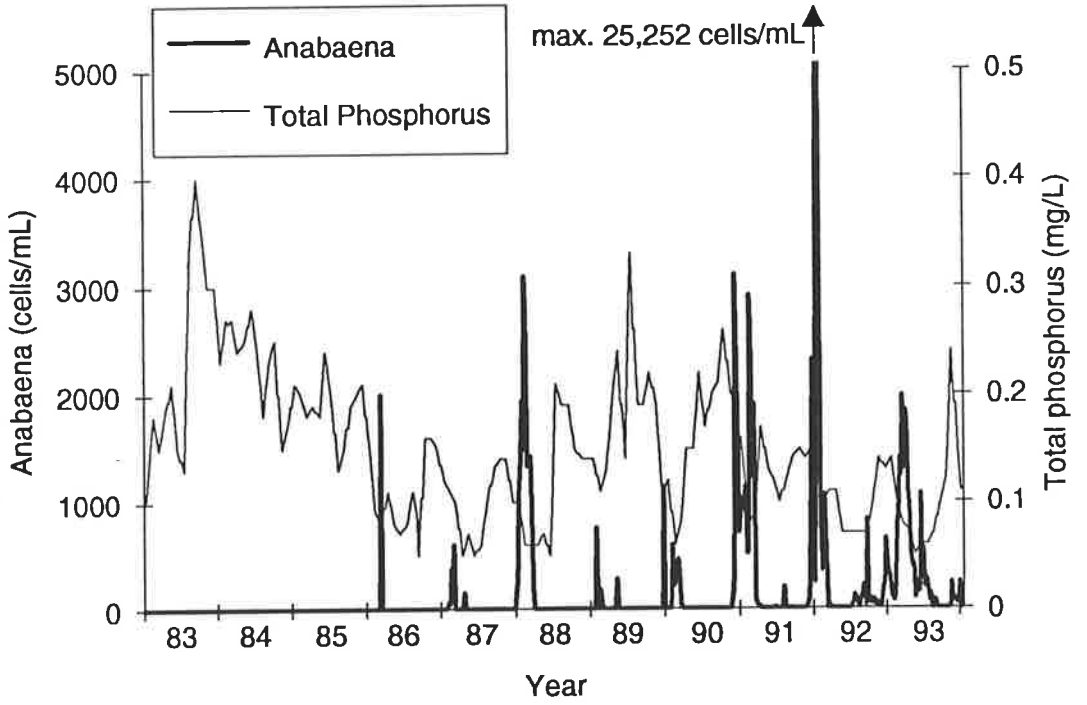


Figure 4.8: Concentrations of *Anabaena* and Total Phosphorus at Morgan (1983 to 1993)

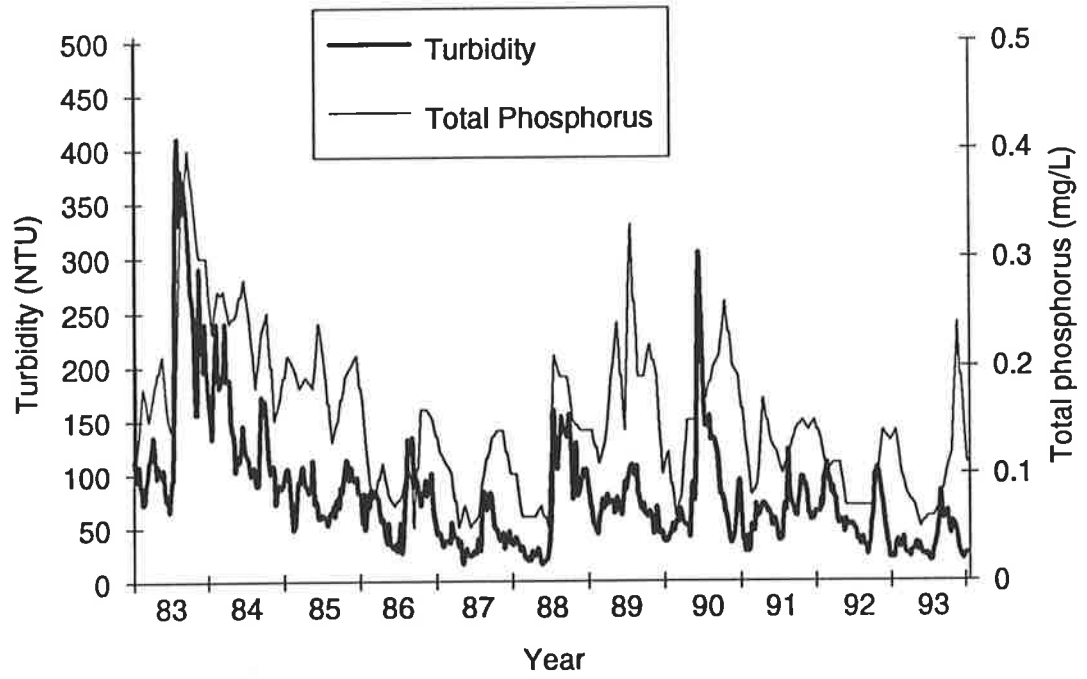


Figure 4.9: Total Phosphorus and Turbidity at Morgan (1983 to 1993)

Soluble phosphorus:

A plot showing the variation of *Anabaena* and soluble phosphorus at Morgan is shown in Figure 4.10. It can be seen that a negative correlation exists between *Anabaena* concentration and soluble phosphorus. This is contrary to what one would expect, as higher levels of nutrients are needed for the growth of algae. However, as discussed in the section on total phosphorus above, high levels of soluble phosphorus and turbidity occur at the same time, but have contradictory effects. High turbidities inhibit algal growth, whereas high levels of soluble phosphorus favour algal growth. Figures 4.5 and 4.10 indicate that soluble phosphorus is not limiting and that turbidity has a significant effect on algal growth rate. The high levels of soluble phosphorus in 1983 and 1984 are due to a high proportion of flow from the Darling River.

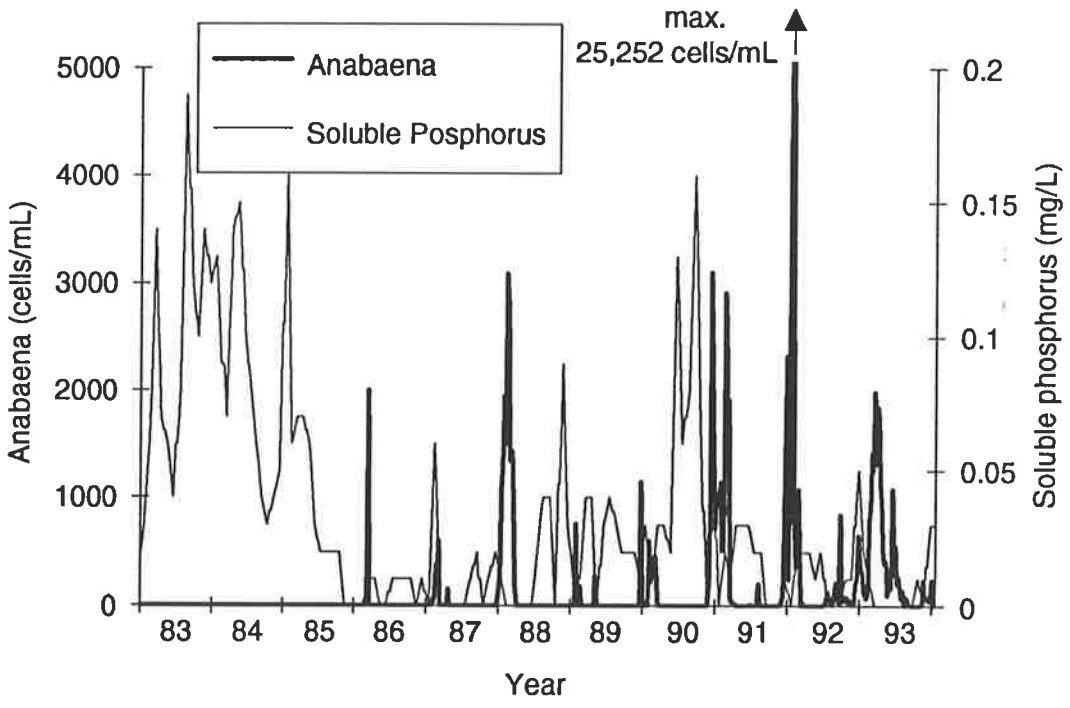


Figure 4.10: Concentrations of *Anabaena* and Soluble Phosphorus at Morgan (1983 to 1993)

Nitrogen:

A plot showing the variation of *Anabaena* and nitrogen at Morgan is shown in Figure 4.11. It can be seen that concentrations of *Anabaena* are inversely proportional to nitrogen, suggesting that nitrogen is not limiting.

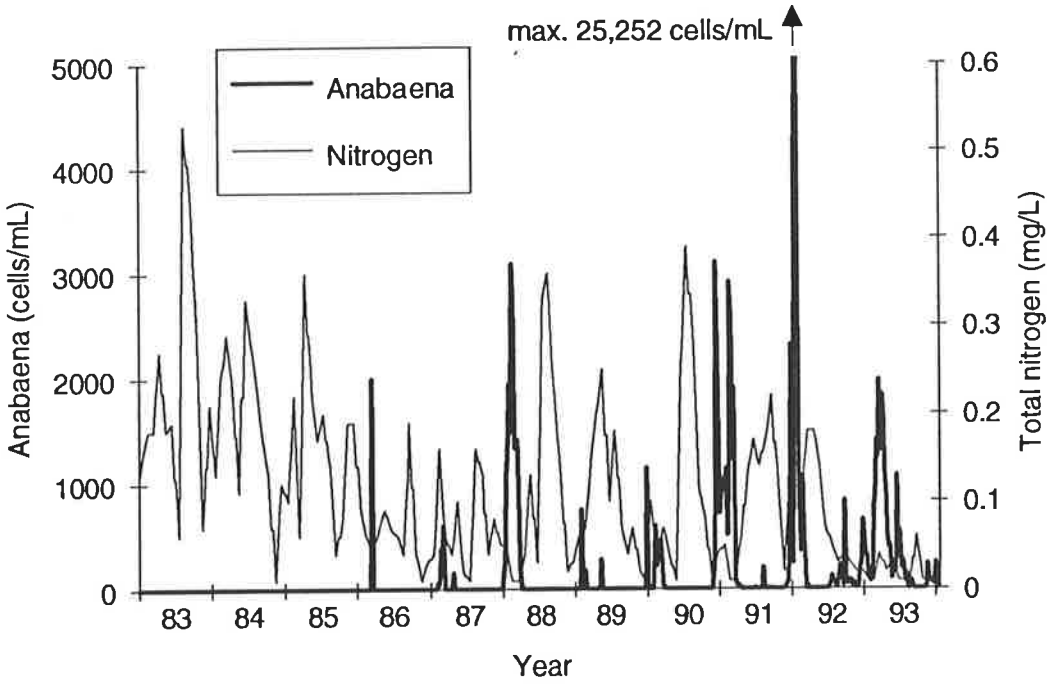


Figure 4.11: Concentrations of *Anabaena* and Nitrogen at Morgan (1983 to 1993)

4.3.2 Implementation

As was the case when the ANN models for forecasting salinity in the River Murray at Murray Bridge were developed, development of the ANN models was carried out on a 80486 PC using the commercially available software package NeuralWare Professional II/Plus (NeuralWare, Inc., 1991). The features of the package are described in Section 3.3.4.1.

4.3.3 Forecasting Period

As discussed previously, obtaining multi-step forecasts is only a secondary objective of the case study considered. Consequently, it was decided to arbitrarily include forecasting periods of 2 and 4 weeks, which would give sufficient warning of impending blooms of *Anabaena*.

4.3.4 Performance Measures

Due to the nature of the time series of *Anabaena* concentrations at Morgan (Figure 4.3), the measures used to assess the performance of the salinity forecasting models (i.e. RMSE, AAPE and AAE) are not appropriate for assessing the performance of models forecasting *Anabaena* concentrations. Figure 4.3 shows that the concentrations of *Anabaena* at Morgan are zero for the majority of the time, with isolated incidences of

Anabaena. As far as water authorities are concerned, the most important criteria for assessing the adequacy of forecasts include the relative magnitude and timing of the incidence of *Anabaena*. However, the models that perform best when performance measures such as the RMSE, AAPE and AAE are used, are not necessarily the models that produce the best forecasts when the criteria of relative magnitude and timing are considered. For example, if there was a large incidence of *Anabaena*, which the model predicted with a slight lag, the RMSE between the actual and predicted values would be very large, and in some instances greater than that obtained if the model predicted no incidence of *Anabaena*. However, the former forecast would be far more useful, especially if the predicted incidence of *Anabaena* occurred slightly before the actual incidence. Consequently, it was decided to adopt relative magnitude and timing of the incidence of *Anabaena* as performance measures.

The predicted and actual peak concentrations of *Anabaena* in any given year were compared in terms of their significance (i.e. significant events have peak concentrations $\geq$ 1000 cells/mL). The timing of the forecasts for the significant events was assessed by visual inspection of the plots of actual versus predicted concentrations.

As no real time forecasts were obtained (see Section 4.1), cross-validation was used as a stopping criterion. The performance of the models was assessed at various stages of learning by inspecting plots of actual versus predicted concentrations of *Anabaena*.

4.3.5 Nomenclature

The following nomenclature will be used in this chapter to identify the different multivariate ANN models developed:

Model Name:	X	X	-	X	X	-	X	X	/	X	X
Cell Number:	1	2	3	4	5	6	7	8	9	10	11

Cell Numbers:	Meaning:
1-2	Model number (01 to 99). Identify internal network parameters (e.g. learning rate, momentum, learning rule, epoch size, error function, transfer function) and network geometry (number of hidden layers and number of nodes per hidden layer). Parameters and geometries corresponding to each model number are given in Appendix D.

4-5 Training / testing set (01 to 99).
Identify network inputs and outputs. The inputs and outputs corresponding to each training / testing set are given in Appendix C.

7-11 Identify the year of data used for forecasting.
(eg. 83/84 = 1983/1984
91/92 = 1991/1992).

4.4 Development of Multivariate Neural Network Model

In this section, the development of the multivariate ANN models for the prediction of *Anabaena* concentrations in the River Murray at Morgan is described. The procedure set out in Figure 2.18 was followed.

4.4.1 Inspection of Plots of the Time Series

Plots and discussions of the time series are given in Section 4.3.1.1.

4.4.2 Determination of Model Inputs

The neural network based approach, which is described in Section 3.10.2, was used to determine appropriate model inputs.

4.4.2.1 Development of Bivariate Models

Seven bivariate models were developed, relating the output variable (i.e. *Anabaena* concentrations) to lagged inputs of each of the seven input variables. A summary of the networks developed for input identification purposes is given in Table 4.6. It should be noted that the maximum lag (k_{max}) used was 26, as it was assumed that present *Anabaena* concentrations would not be affected by inputs from more than six months earlier.

It was decided to use all 10 years of data (i.e from 1983/84 to 1992/93) for training and to use the same data for testing. This indicates how well the relationships between the input data and the output data have been learnt by the networks. As a result, the nomenclature of the models given in Table 4.6 only shows the model number and the number of the training / testing set, and not the year used for forecasting, as all ten years of data were used for forecasting.

Table 4.6: Details of the Bivariate Models Developed for Input Identification Purposes

Model	Output variable	Input variable	Lags of inputs (weeks)
01_01	*Anabaena spp.*	Colour	1, 2, ...,25, 26
01_02	*Anabaena spp.*	Turbidity	1, 2, ...,25, 26
01_03	*Anabaena spp.*	Temperature	1, 2, ...,25, 26
02_04	*Anabaena spp.*	Flow	1, 2, ...,25, 26
01_05	*Anabaena spp.*	Soluble Phosphorus	1, 2, ...,25, 26
02_06	*Anabaena spp.*	Total Phosphorus	1, 2, ...,25, 26
01_07	*Anabaena spp.*	Nitrogen	1, 2, ...,25, 26

The network geometry was chosen to be 26-45-15-1 for all networks. It was decided to use two hidden layers and a relatively large number of nodes per layer to ensure that sufficient parameters were available to estimate the complex relationship between the network inputs and outputs. It should be noted that overtraining is not an issue in this case, as the training set is used for testing, and consequently, the generalisation ability of the networks is not tested. The lag of the output was chosen to be +1 (i.e. a 2-week forecast).

Initially, the same internal parameters were used for all seven models. Based on the results obtained in Chapter 3, a learning rate of 0.1, a momentum value of 0.6, an epoch size of 16, the hyperbolic tangent transfer function, the quadratic error function and the normalised cumulative delta learning rule were used. However, when flow and total phosphorus were used as input variables, the steps taken in weight space were too large and learning ceased. As a result, the learning rate was reduced from 0.1 to 0.02, the connection weights were re-initialised and training was re-commenced.

The testing interval was chosen to be 25,000. The plots of the forecasts obtained did not change substantially after the presentation of 150,000 training samples, and training was ceased at a learn count of 200,000 for all networks.

The plots of the forecasts obtained using all seven models, after the presentation of 200,000 training samples, are shown in Figures 4.12 to 4.18. It is interesting to note that despite the fact that the same data were used for training and testing, and that large network geometries were used, the models were only able to predict certain incidences of *Anabaena*. In addition, different events were predicted by different variables. The relative magnitude of the actual events and the events predicted by the various models

for the ten years of data are summarised in Table 4.7. It should be noted that the results for the model using temperature inputs have been omitted from Table 4.7, as this model predicts a peak of approximately the same size at the same time each year (Figure 4.14).

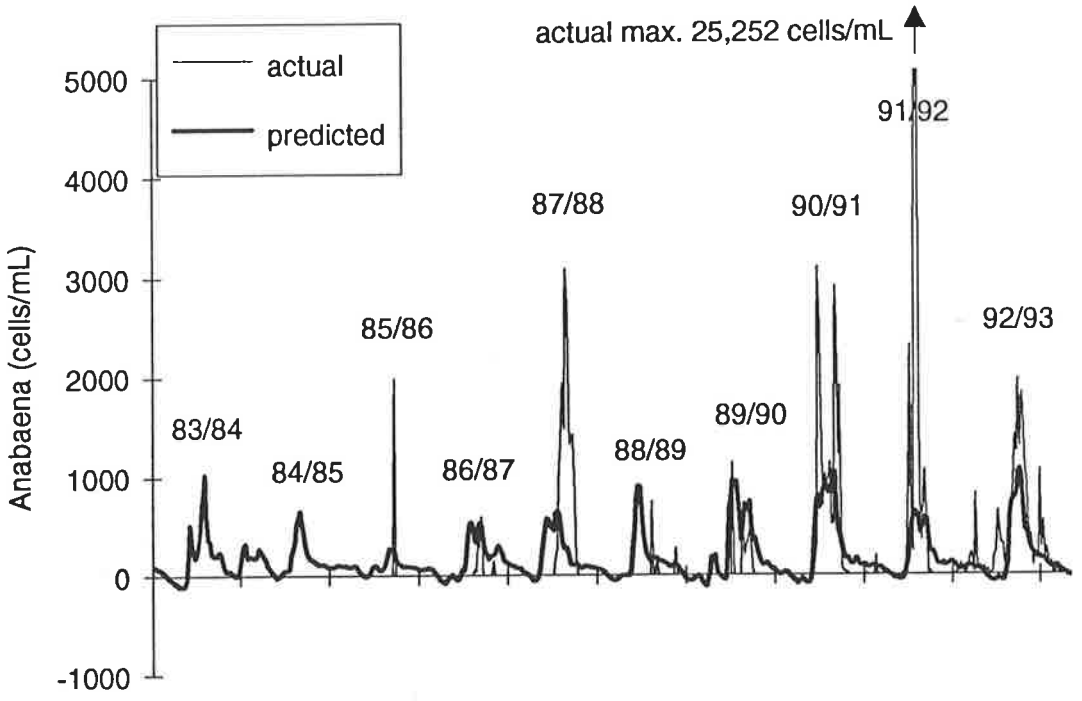


Figure 4.12: Two-Week Forecasts Obtained Using the Model With Colour Inputs (Model 01-01)

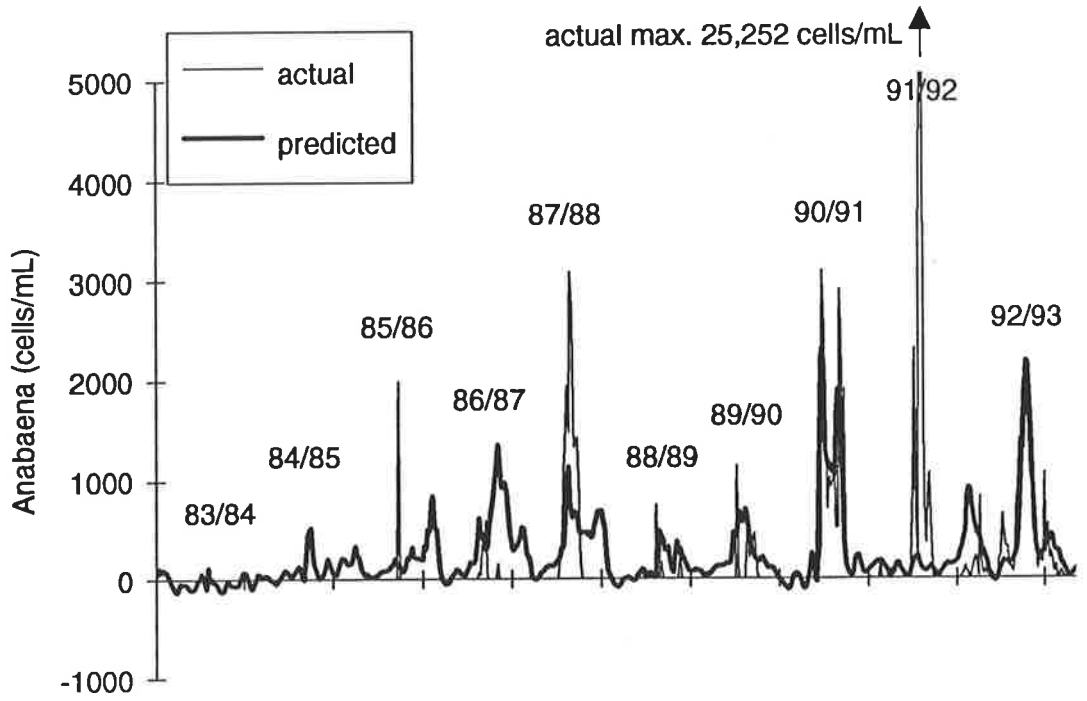


Figure 4.13: Two-Week Forecasts Obtained Using the Model With Turbidity Inputs (Model 01-02)

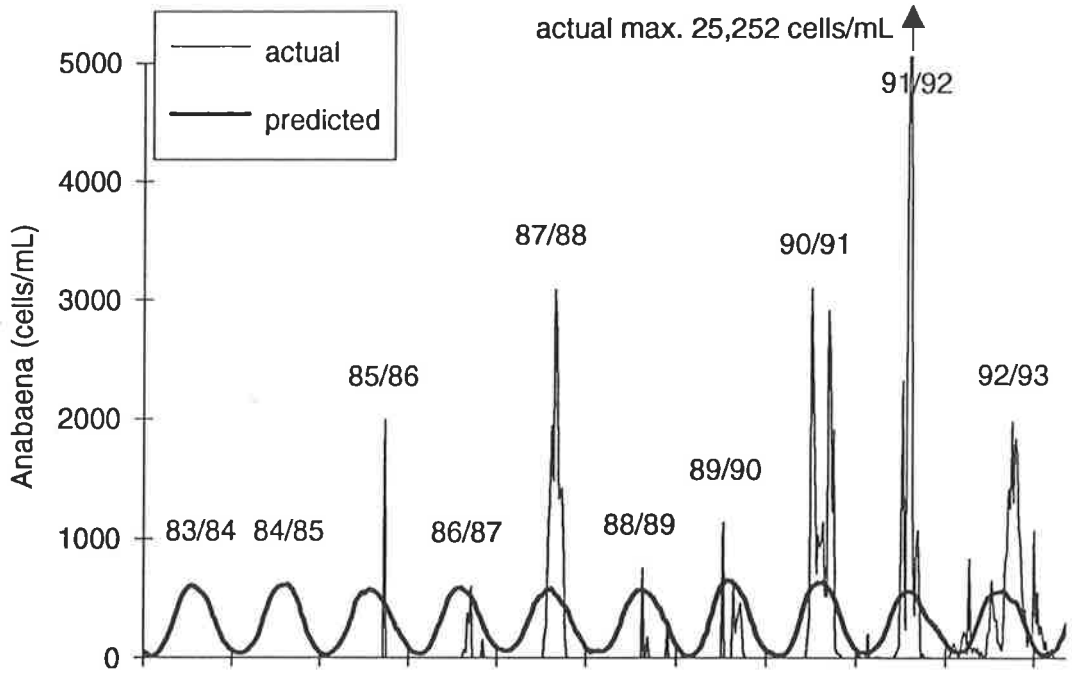


Figure 4.14: Two-Week Forecasts Obtained Using the Model With Temperature Inputs (Model 01-03)

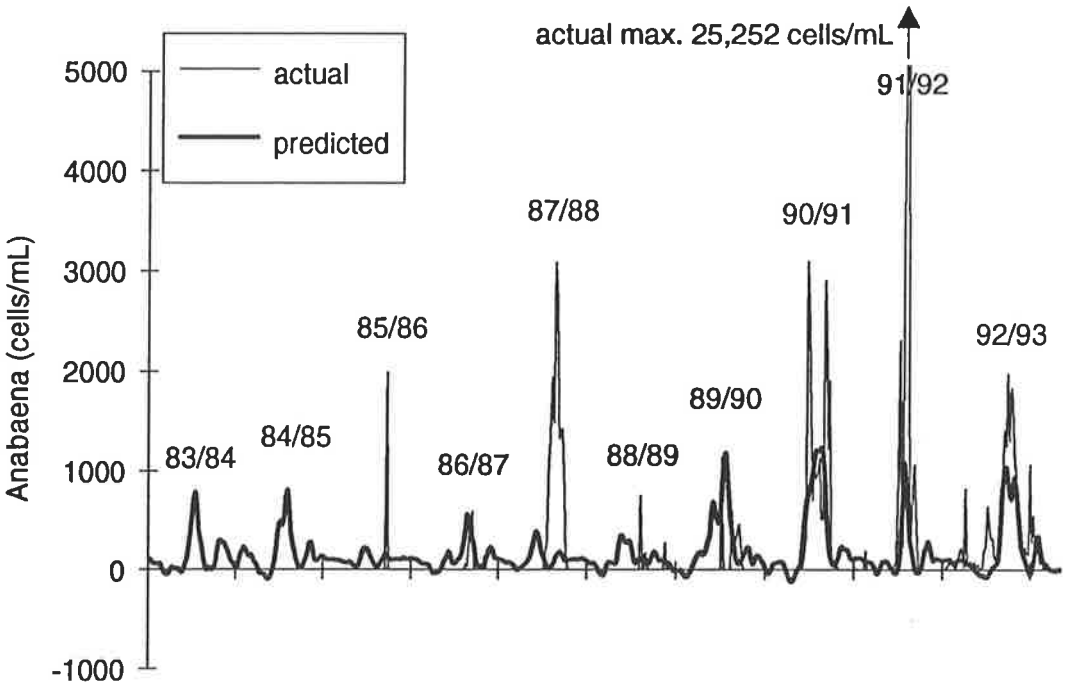


Figure 4.15: Two-Week Forecasts Obtained Using the Model With Flow Inputs (Model 02-04)

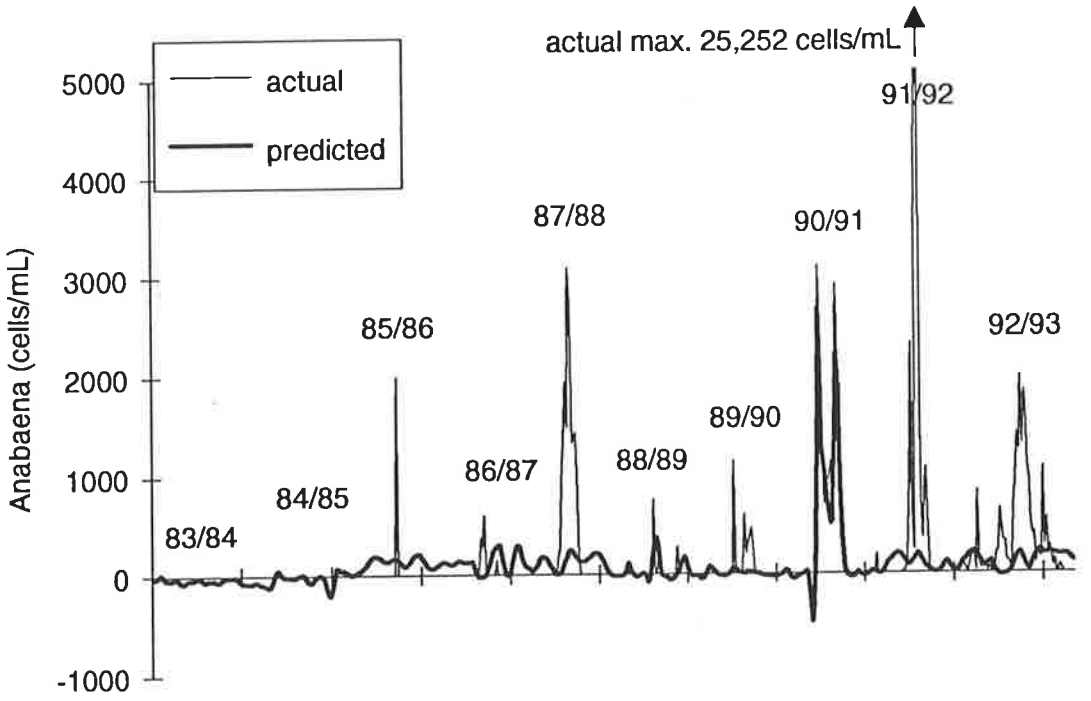


Figure 4.16: Two-Week Forecasts Obtained Using the Model With Soluble Phosphorus Inputs (Model 01-05)

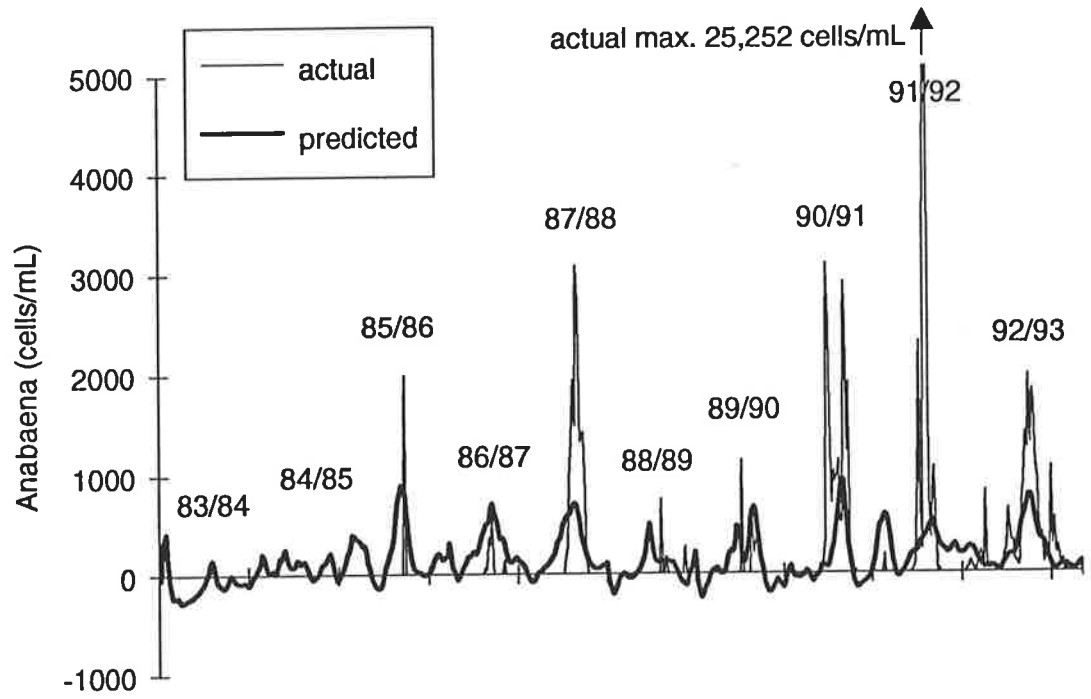


Figure 4.17: Two-Week Forecasts Obtained Using the Model With Total Phosphorus Inputs (Model 02-06)

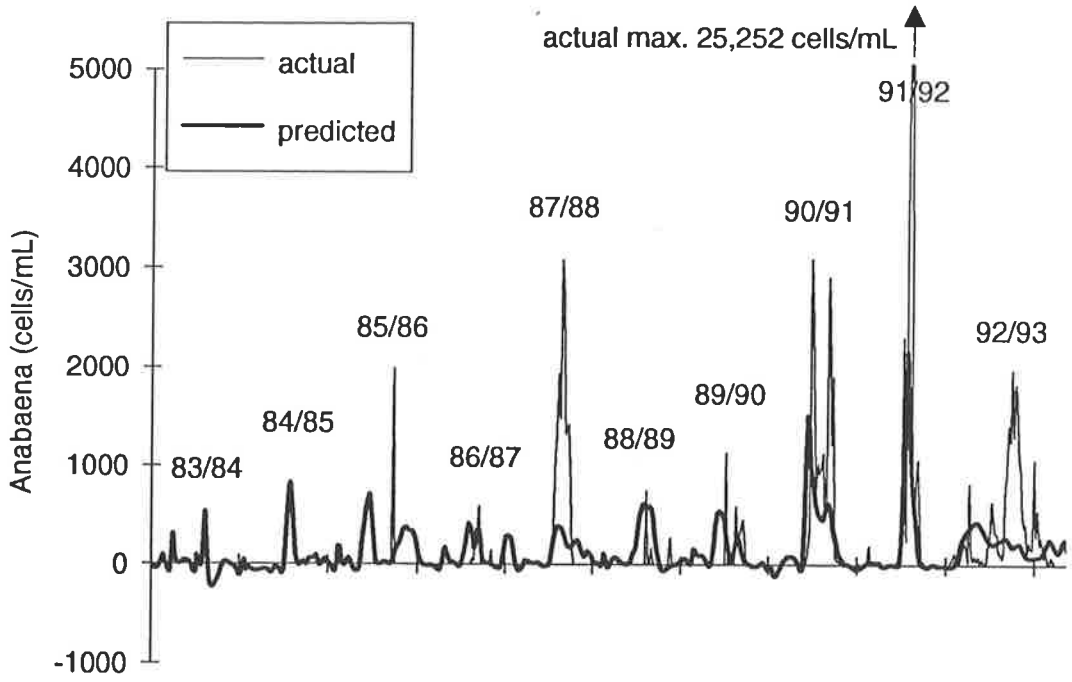


Figure 4.18: Two-Week Forecasts Obtained Using the Model With Nitrogen Inputs (Model 01-07)

If one of the models correctly predicts the incidence or absence of *Anabaena* in a particular year, one can conclude that some relationship exists between the input variable considered (i.e colour, turbidity etc.) and concentrations of *Anabaena* (in that particular year). However, the relationship obtained is not necessarily general, as the same data were used for training and testing. If, on the other hand, one of the models does not correctly predict the incidence or absence of *Anabaena* in a particular year, despite the fact that the same data were used for training and testing, one can conclude that there is definitely no relationship between the variable considered and concentrations of *Anabaena*.

Table 4.7 and Figures 4.12 to 4.18 indicate that:

- The lack of any incidence of *Anabaena* in 1983/84 and 1984/85 was only predicted by the models using turbidity, soluble phosphorus and total phosphorus as inputs. This tends to suggest that the high levels of these three variables in the years in question are responsible for the lack of *Anabaena*. However, as discussed previously, turbidity, soluble phosphorus and total phosphorus are highly correlated, and high levels of turbidity inhibit the growth of *Anabaena*, whereas high levels of phosphorus favour their growth. Consequently, one can conclude that high levels of turbidity were probably responsible for the absence of Anabaena in 1983/84 and

1984/85. The fact that the models using colour, temperature, flow and nitrogen as inputs predicted incidence of *Anabaena*, when in fact there were none, indicates that colour, temperature, flow and nitrogen were definitely not responsible for the lack of *Anabaena* in 1983/84 and 1984/85.

Table 4.7: Relative Magnitude of Actual and Predicted Peak Concentrations of *Anabaena*

Year	Actual	Predicted					
		Input variable					
		Colour	Turbidity	Flow	Soluble phosphorus	Total phosphorus	Nitrogen
83/84		++		+			+
84/85		+		+			+
85/86	++					++	
86/87	+	+	++	+		+	+
87/88	+++	+	++			+	+
88/89	+	+	+		+	+	+
89/90	++	+	+	++		+	+
90/91	+++	++	+++	++	+++	++	++
91/92	+++++	+		++		+	+++
92/93	++	++	+++	++		+	

0 cells/mL ≤ concentration of *Anabaena* < 500 cells/mL

\+ 500 cells/mL ≤ concentration of *Anabaena* < 1000 cells/mL

++ 1000 cells/mL ≤ concentration of *Anabaena* < 2000 cells/mL

+++ 2000 cells/mL ≤ concentration of *Anabaena* < 3000 cells/mL

++++ 4000 cells/mL ≤ concentration of *Anabaena* < 5000 cells/mL

+++++ concentration of *Anabaena* ≥ 5000 cells/mL

- The sharp peak in *Anabaena* concentration in 1985/86 was only predicted by the model using total phosphorus as inputs, suggesting that total phosphorus might be responsible for this incidence of *Anabaena*. The fact that the other models were unable to predict the peak in *Anabaena* concentration in 1985/86 indicates that colour, turbidity, temperature, flow, soluble phosphorus and nitrogen were definitely not responsible for the above peak in *Anabaena* concentration.

- The small incidence of *Anabaena* in 1986/87 was predicted by all models, apart from the one using soluble phosphorus as inputs, suggesting that soluble phosphorus was definitely not responsible for the incidence of *Anabaena* in 1986/87. The incidence of *Anabaena* predicted by the model using turbidity inputs was much larger than the actual incidence.

- The large incidence of *Anabaena* in 1987/88 was predicted by the models using colour, turbidity, total phosphorus and nitrogen as inputs, although the timing of the incidence predicted by the model using colour inputs, and the magnitude of the incidence predicted by the model using nitrogen inputs, were not very good. The models using flow and soluble phosphorus as inputs did not predict the incidence of *Anabaena* in 1987/88, suggesting the these two variables were definitely not responsible for this incidence of *Anabaena*.

- The small incidence of *Anabaena* in 1988/89 was predicted by all models, apart from the one using flow inputs, suggesting that flow was definitely not responsible for the incidence of *Anabaena* in 1988/89.

- The incidence of *Anabaena* in 1989/90 was predicted by all models, apart from the one using soluble phosphorus as inputs, suggesting that soluble phosphorus was definitely not responsible for the incidence of *Anabaena* in 1989/90.

- The large incidence of *Anabaena* in 1990/91 was predicted by all models, although the double peak in the concentration of *Anabaena* was only predicted by the models using turbidity and soluble phosphorus as inputs.

- The very large incidence of *Anabaena* in 1991/92 was predicted by the models using colour, flow, total phosphorus and nitrogen as inputs, although the predictions obtained using the models with flow and nitrogen inputs were by far the best. The models using turbidity and soluble phosphorus as inputs did not predict the incidence of *Anabaena* in 1991/92, suggesting the these two variables were definitely not responsible for this incidence of *Anabaena*.

- The large incidence of *Anabaena* in 1992/93 was predicted by the models using colour, turbidity, flow and total phosphorus as inputs. The models using soluble phosphorus and nitrogen as inputs did not predict the incidence of *Anabaena* in 1992/93, suggesting the these two variables were definitely not responsible for this incidence of *Anabaena*.

The above results indicate that the relationship between concentrations of *Anabaena* and the prevailing environmental conditions is very complex, as different variables appear to be responsible for the incidence of *Anabaena* in different years.

4.4.2.2 Performance of Sensitivity Analyses

As part of the sensitivity analyses, each of the network inputs was increased by an amount equal to 5% of the average value of the input variable under consideration, which was calculated by considering the ten years of data available. The resulting change in the predicted algal concentration was recorded. The amount by which each of the inputs was increased is given in Table 4.8 for the various input variables.

Table 4.8: Amount by Which Each of the Inputs was Increased During the Sensitivity Analyses

Input Variable	Increase for each Input
Colour	1.94 HU
Turbidity	4.25 NTU
Temperature	0.94 °C
Flow	1178 ML/Day
Total Phosphorus	0.008 mg/L
Soluble Phosphorus	0.002 mg/L
Nitrogen	0.006 mg/L

Typical plots of the relative significance of the inputs of each of the seven bivariate models are shown in Figures 4.19 to 4.25. It should be noted that the sensitivities shown are those at the times of maximum *Anabaena* concentrations in any given year. The sensitivities for different years are shown, as the relative significance of the inputs varied from year to year. Only the plots of the relative significance of the inputs for the years for which the models accurately predicted the incidence of *Anabaena* are shown. As there were no *Anabaena* present in 1983/84 and 1984/85, no plots are shown for these years.

As can be seen from Figures 4.19 to 4.25, the cut-off points between significant and non-significant inputs are not as clearly defined as they were in the salinity forecasting problem. A great deal of judgement had to be used to determine which inputs to include in the multivariate model. In many instances, there appear to be two cut-off points, as

indicated by the vertical lines in Figures 4.19 to 4.25. For example, in Figure 4.20, there appears to be a cut-off point at lag 8 and again at lag 22. It was decided to compile two different training / testing sets; one which included inputs up to the first cut-off point, and a second which included inputs up to the second cut-off point. It should be noted that in this case study, the choice of which inputs to include in the multivariate model requires a great deal of judgement. The lags of the inputs chosen for the various input time series using this method are given in Table 4.9 (training / testing sets 09 and 10). In order to assess the effect of using lagged inputs, a third training / testing set was also used, which included inputs at lag 1 only (training / testing set 08, Table 4.9).

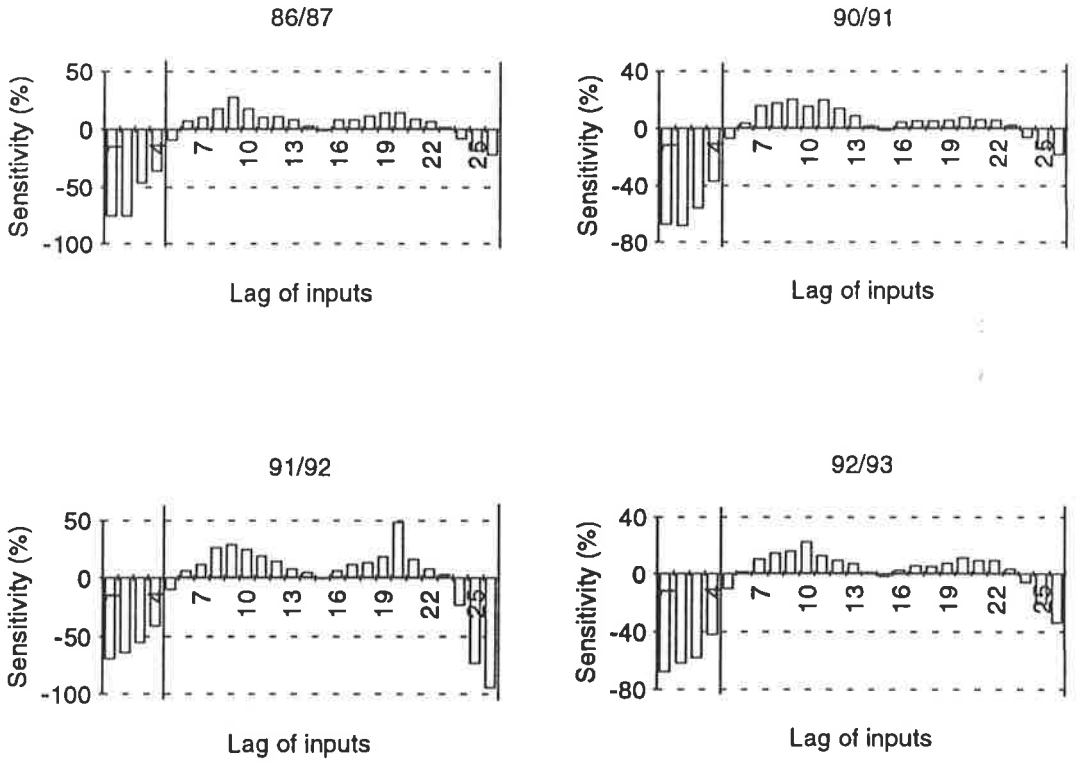


Figure 4.19: Relative Significance of Inputs - Model 01_01 (i.e. Model Using Colour Inputs)

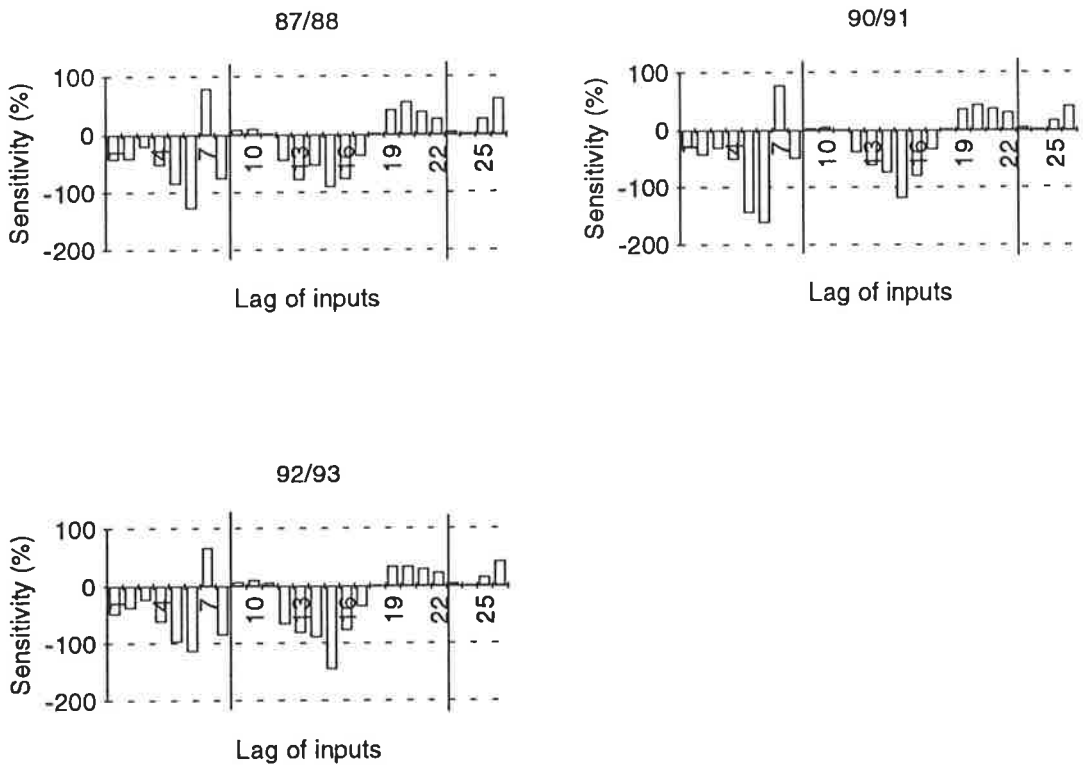


Figure 4.20: Relative Significance of Inputs - Model 01_02 (i.e. Model Using Turbidity Inputs)

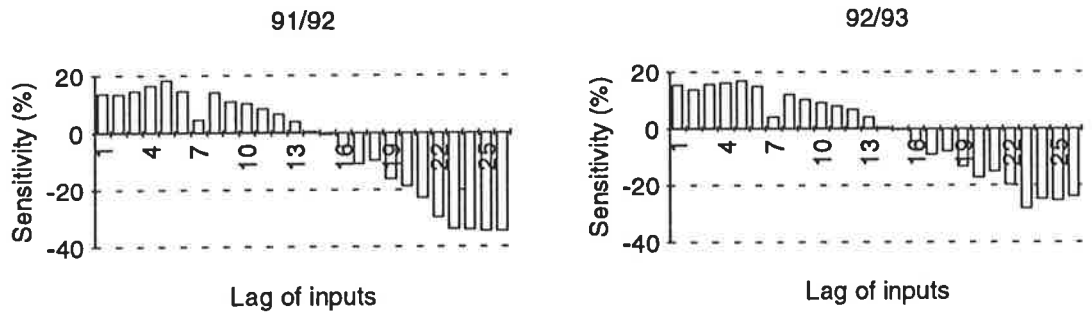


Figure 4.21: Relative Significance of Inputs - Model 01_03 (i.e. Model Using Temperature Inputs)

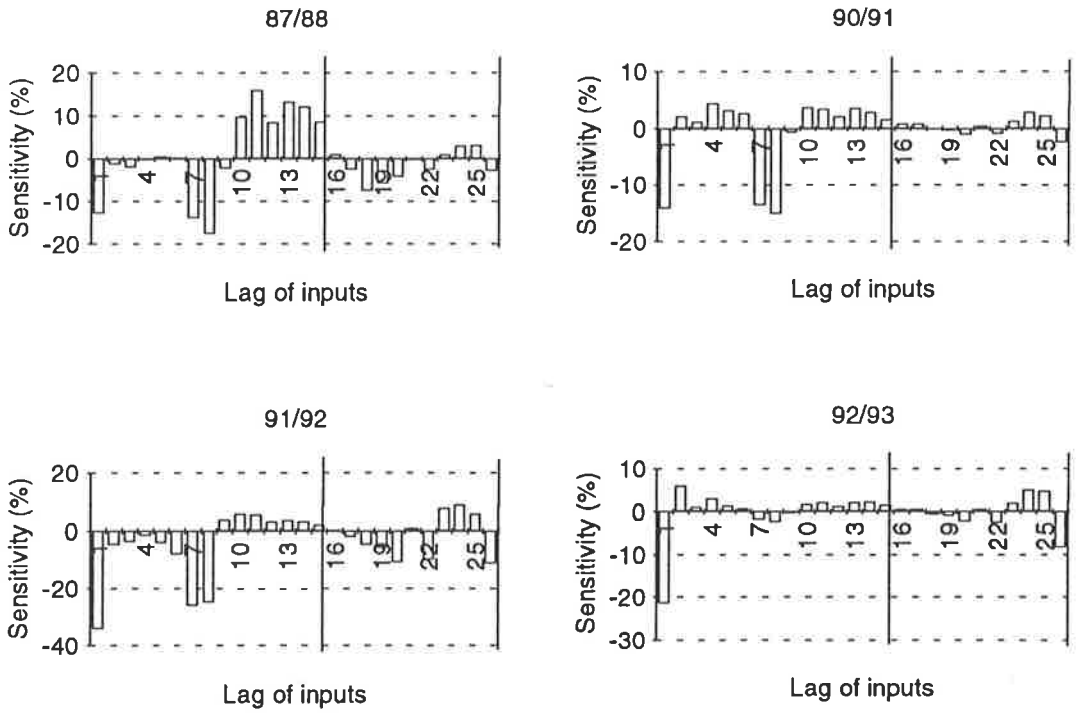


Figure 4.22: Relative Significance of Inputs - Model 02_04 (i.e. Model Using Flow Inputs)

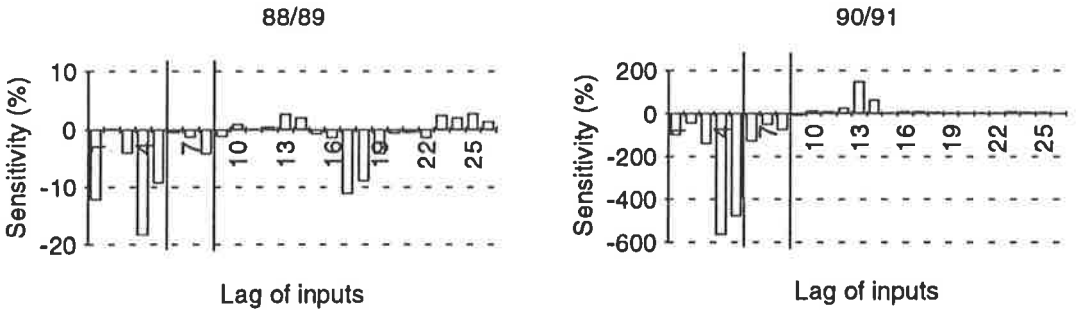


Figure 4.23: Relative Significance of Inputs - Model 01_05 (i.e. Model Using Soluble Phosphorus Inputs)

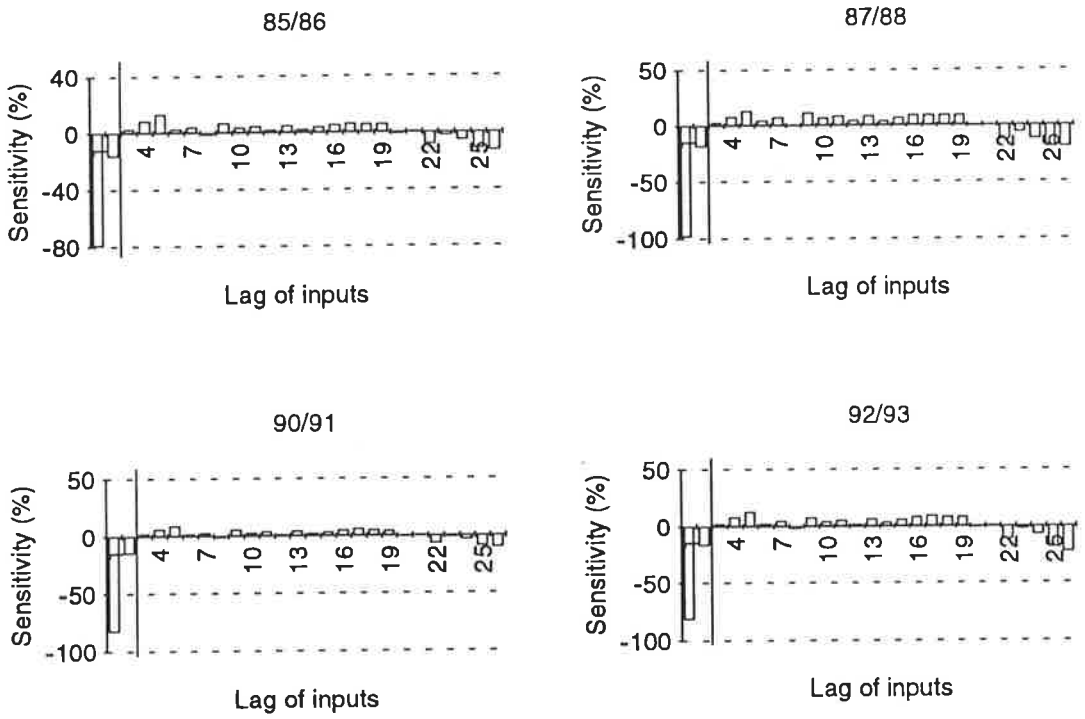


Figure 4.24: Relative Significance of Inputs - Model 02_06 (i.e. Model Using Total Phosphorus Inputs)

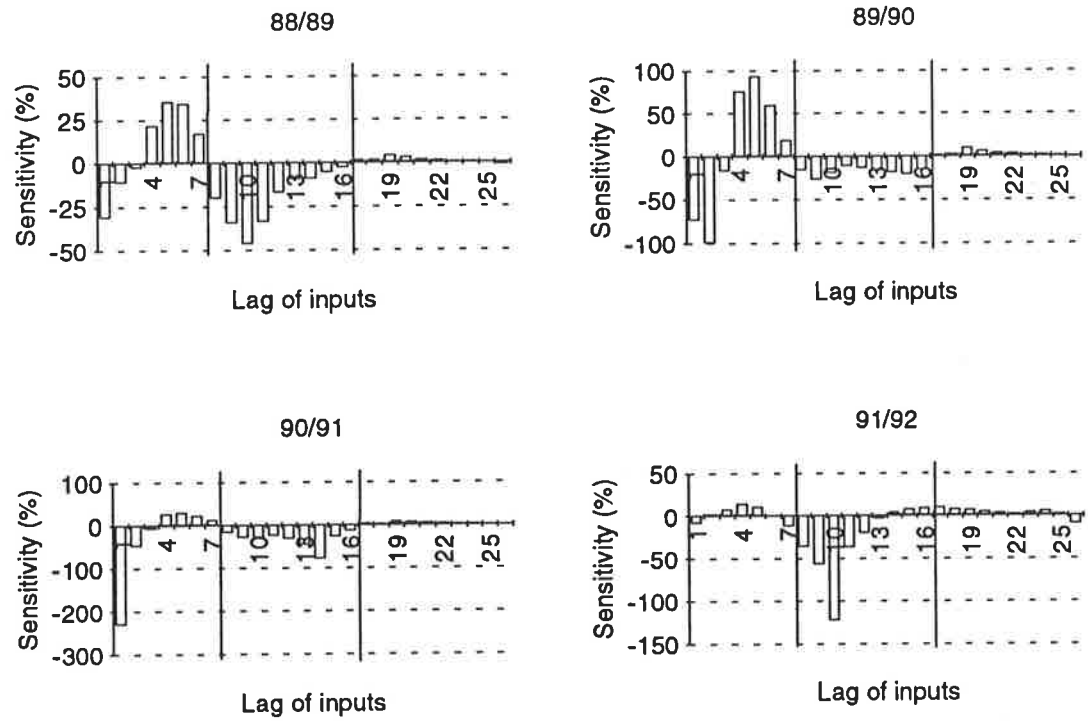


Figure 4.25: Relative Significance of Inputs - Model 01_07 (i.e. Model Using Nitrogen Inputs)

Table 4.9: Multivariate Model Inputs Chosen

Input variable	Lags of inputs		
	Training / testing set 08	Training / testing set 09	Training / testing set 10
Colour	1	1, 2, 3, 4	1, 2, ..., 26
Turbidity	1	1, 2, ..., 8	1, 2, ..., 22
Temperature	1	1, 26	1, 26
Flow	1	1, 2, ..., 15	1, 2, ..., 26
Soluble phosphorus	1	1, 2, ..., 5	1, 2, ..., 8
Total phosphorus	1	1, 2	1, 2
Nitrogen	1	1, 2, ..., 7	1, 2, ..., 16
	7 Total	43 Total	102 Total

4.4.3 Choice of Network Geometry and Internal Parameters

As a result of the complexity of the relationship between the input and output variables, it was decided to use two hidden layers. Based on the results obtained in Chapter 3, the network geometries for training / testing sets 08, 09 and 10 were chosen to be 7-15-5-2, 43-45-15-2 and 102-120-40-2 respectively. As discussed in Section 4.3.3, the lags of the outputs were chosen to be +1 (i.e. 2-week forecast) and +3 (i.e. 4-week forecast).

The results obtained in Chapter 3 were also used as a guide for choosing the internal network parameters. A learning rate of 0.03, a momentum of 0.6, an epoch size of 16, the hyperbolic tangent transfer function and the normalised cumulative delta learning rule were used.

4.4.4 Training

In order to obtain good model performance, the training data were chosen so that they contained all extreme values of the input and output variables. As mentioned in Section 4.3.1.1, the turbidities in 1983/84 and the *Anabaena* concentrations in 1991/92 were significantly greater than the values in any other years. Consequently, data from these years were not used for validation purposes.

In order to make maximum use of the available data, and to get the best possible indication of the generalisation ability of the models, eight networks were trained using

each of the three training / testing sets (i.e. training / testing sets 08, 09 and 10), as shown in Table 4.10. Each of the eight networks were trained using all available data with the exception of one year, which was withheld to obtain an independent forecast (Table 4.10). The data from 1984/85, 1985/86, 1986/87, 1987/88, 1988/89, 1989/90, 1990/91 and 1992/93 were withheld in turn. The data from 1983/84 and 1991/92 were not used for forecasting for reasons discussed above. This method enables independent forecasts to be obtained for a number of years, while using the maximum amount of data available for training.

The testing interval was chosen to be 5,000. Using cross-validation as the stopping criterion, training was ceased at a learn count of 50,000. The models which produced the best forecasts were retained.

Table 4.10: Multivariate Models Trained for Each of the 3 Training / Testing Sets

Model			Years used for training	Year used for testing
Training / testing set				
08 (7 inputs)	09 (43 inputs)	10 (102 inputs)		
03_08_84/85	04_09_84/85	05_10_84/85	All but 1984/85	1984/85
03_08_85/86	04_09_85/86	05_10_85/86	All but 1985/86	1985/86
03_08_86/87	04_09_86/87	05_10_86/87	All but 1986/87	1986/87
03_08_87/88	04_09_87/88	05_10_87/88	All but 1987/88	1987/88
03_08_88/89	04_09_88/89	05_10_88/89	All but 1988/89	1988/89
03_08_89/90	04_09_89/90	05_10_89/90	All but 1989/90	1989/90
03_08_90/91	04_09_90/91	05_10_90/91	All but 1990/91	1990/91
03_08_92/93	04_09_92/93	05_10_92/93	All but 1992/93	1992/93

4.4.5 Results / Discussion

The ability of the models with 7, 43 and 102 inputs to accurately forecast the relative magnitude of concentrations of *Anabaena* at Morgan is shown in Table 4.11.

It can be seen that model performance increases with an increase in the number of model inputs. All models accurately predict the events with concentrations below 1000 cells/mL in 1984/85, 1986/87 and 1988/89. However, only the model using 102 inputs predicts all 5 events in excess of 1000 cells/mL in 1985/86, 1987/88, 1989/90, 1990/91 and 1992/93. The model using 7 inputs does not predict the significant events in 1987/88 and 1990/91, while the model using 43 inputs does not predict the significant

event in 1987/88. The models which were unable to predict significant incidence of *Anabaena* (models 03_08_87/88, 03_08_90/91 and 04_09_87/88) were re-trained using a larger learning rate of 0.3 (models 06_08_87/88, 06_08_90/91 and 07_09_87/88), in case the models with the smaller learning rates had got stuck in a local minimum in the error surface which they could not escape. However, using a larger learning rate had no significant effect on the forecasts obtained.

Table 4.11: Relative Magnitude of 2-Week Forecasts

Year	Actual peak concentration (cells/mL)	Correct (✓) or incorrect (✗) relative magnitude of forecasts		
		7 I/P	43 I/P	102 I/P
1984/85	< 1000	✓	✓	✓
1985/86	≥ 1000	✓	✓	✓
1986/87	< 1000	✓	✓	✓
1987/88	≥ 1000	✗	✗	✓
1988/89	< 1000	✓	✓	✓
1989/90	≥ 1000	✓ *)	✓	✓
1990/91	≥ 1000	✗	✓	✓
1992/93	≥ 1000	✓	✓	✓

*) Forecast very close to 1000 cells/mL (just below)

In addition to relative magnitude, the timing of the significant events is also of great importance. The timing of the forecasts was assessed by visual inspection of the plots of actual versus predicted concentrations (Figures 4.26 to 4.28). The model with 102 inputs correctly predicts the timing of 4 of the 5 significant events (1987/88, 1989/90, 1990/91 and 1992/93), the model with 43 inputs correctly predicts the timing of 3 of the 5 significant events (1989/90, 1990/91 and 1992/93), while the model with 7 inputs correctly predicts the timing of only 2 of the 5 significant events (1989/90 and 1992/93) (Table 4.12).

The fact that the model with 102 inputs performs best confirms that ANNs are relatively insensitive to model noise and make use of any additional information presented to them. This is especially true for complex problems, such as the one considered in this case study. The results obtained also indicate that the generalisation ability of the models was not adversely affected by using a large number of inputs, and hence large network geometries, despite the fact that only limited training data were available. This is in agreement with the results obtained in Chapter 3.

Table 4.12: Timing of 2-Week Forecasts

Year	Correct (✓) or incorrect (✗) timing of forecasts		
	7 I/P	43 I/P	102 I/P
1985/86	✗	✗	✓
1987/88	N/A	N/A	✗
1989/90	✓	✓	✓
1990/91	N/A	✓	✓
1992/93	✓	✓	✓

Figures 4.26 to 4.28 indicate that the model with 102 inputs has the ability to determine more detailed variations in *Anabaena* concentration than the models with 7 and 43 inputs, such as the double peaks in 1986/87, 1989/90 and 1992/93.

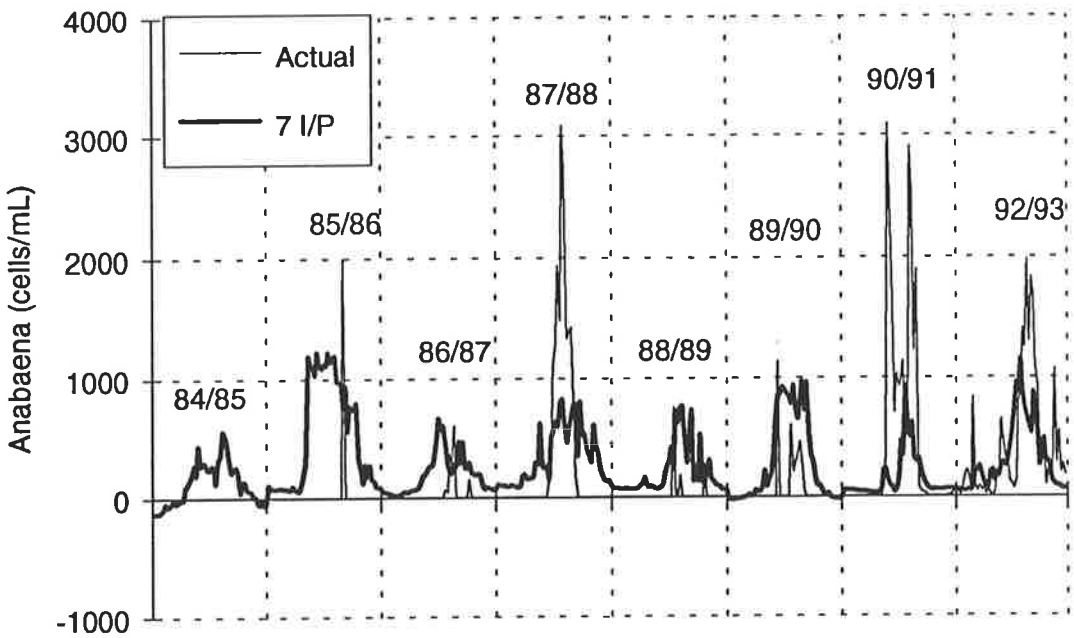


Figure 4.26: Actual and Predicted Concentrations of *Anabaena* at Morgan (2-Week Forecast) - Model With 7 Inputs

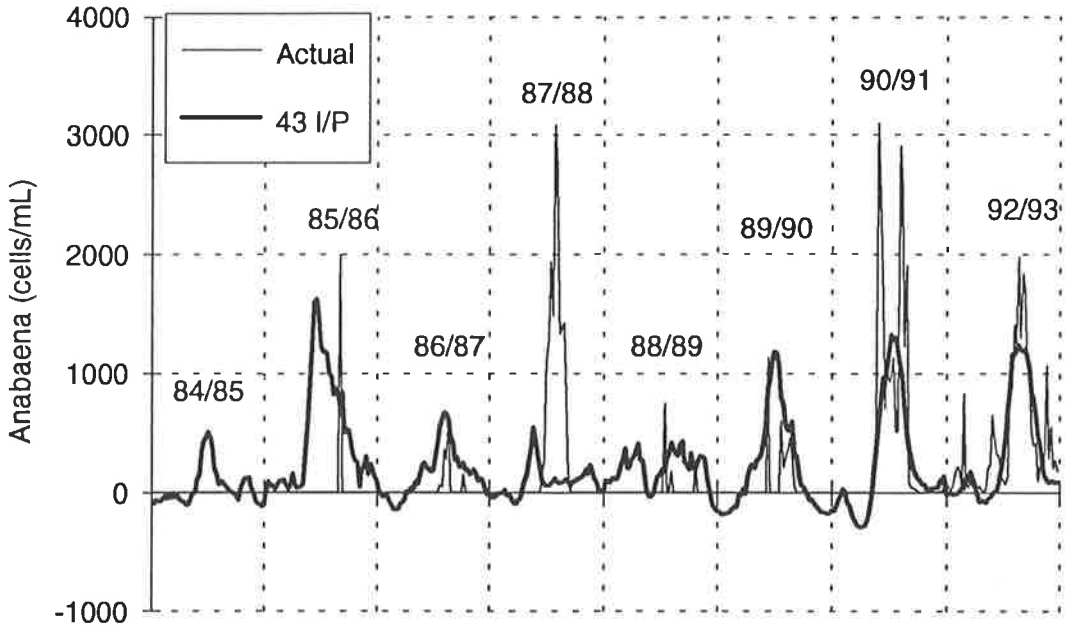


Figure 4.27: Actual and Predicted Concentrations of *Anabaena* at Morgan (2-Week Forecast) - Model With 43 Inputs

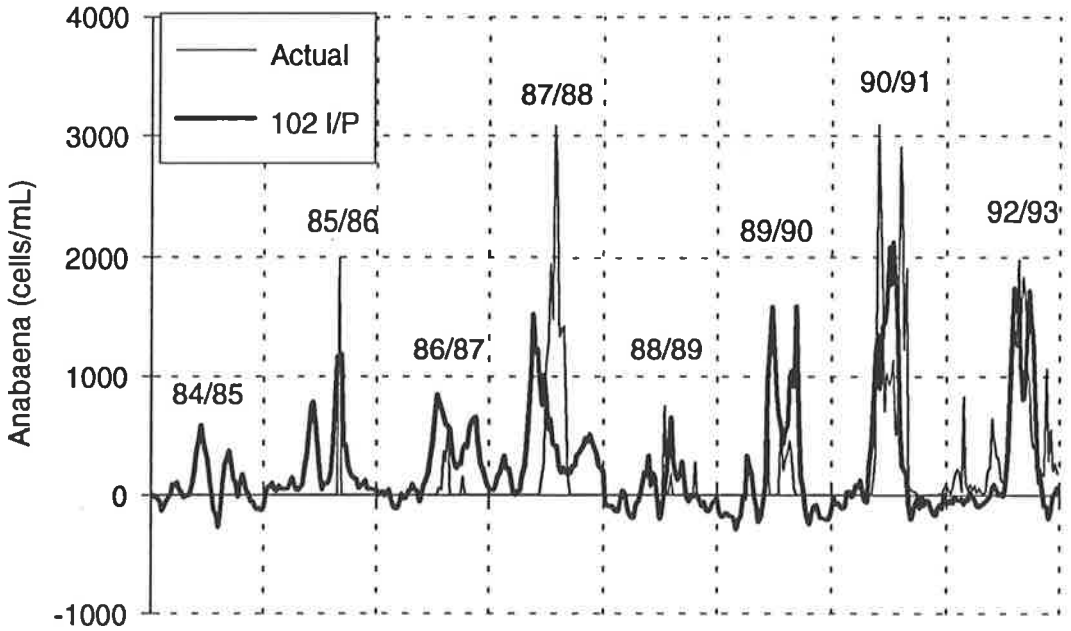


Figure 4.28: Actual and Predicted Concentrations of *Anabaena* at Morgan (2-Week Forecast) - Model With 102 Inputs

A plot of the 4-week forecasts obtained using the model with 102 inputs is shown in Figure 4.29. As can be seen by comparing Figures 4.28 and 4.29, the 4-week forecasts are very similar to the 2-week forecasts. All significant incidence of *Anabaena* are predicted 4 weeks in advance, and the timing of 4 of the 5 significant events is predicted correctly, as is the case for the 2-week forecasts.

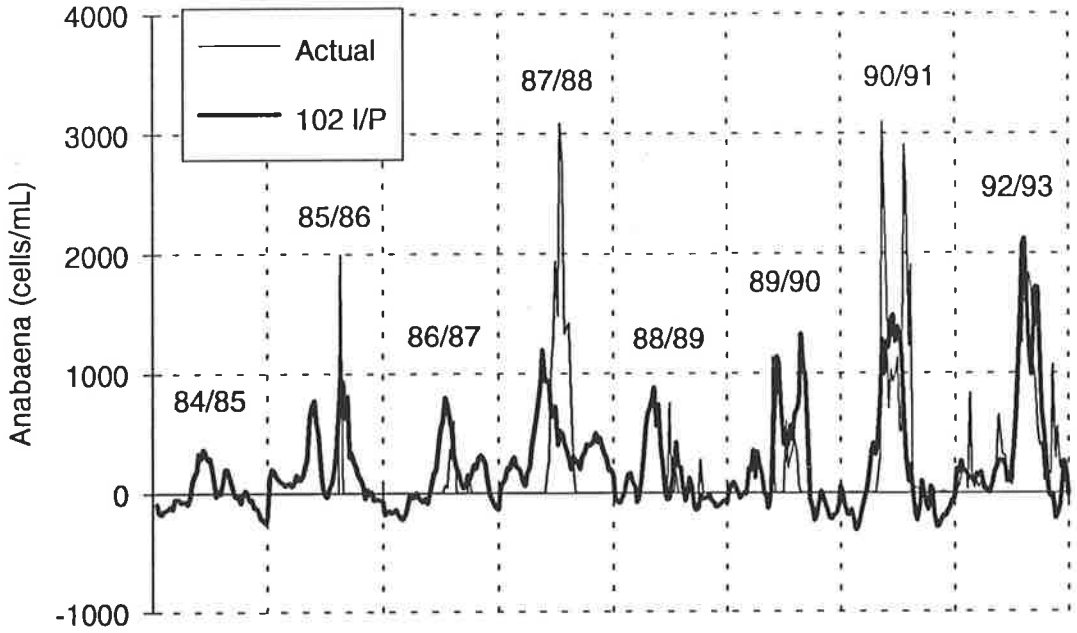


Figure 4.29: Actual and Predicted Concentrations of *Anabaena* at Morgan (4-Week Forecast) - Model With 102 Inputs

A sensitivity analysis was carried out on the model with 102 inputs, in order to determine the relative significance of each of the inputs, and to get some understanding of which environmental conditions are dominant. A typical plot of the relative significance of each of the 102 inputs for a forecasting period of 2 weeks is shown in Figure 4.30. It should be noted that Figure 4.30 is typical for all years used for forecasting, which suggests that a general relationship between the inputs and outputs has been determined by the ANN model. Figure 4.30 indicates that the relationship between the model inputs and outputs is very complex. None of the input variables appears to be dominant, with each one contributing in an approximately equal fashion.

As expected, the sensitivities indicate a strong inverse relationship between algal concentration and turbidity and colour, and a strong positive correlation between temperature and the size of the algal population. The sensitivities also show a strong positive correlation between the concentration of *Anabaena* and flow at lags in excess

of several weeks. This indicates that large algal populations occur after the passing of a flood, which could be responsible for flushing algal cells from anabranches and lagoons into the main river channel.

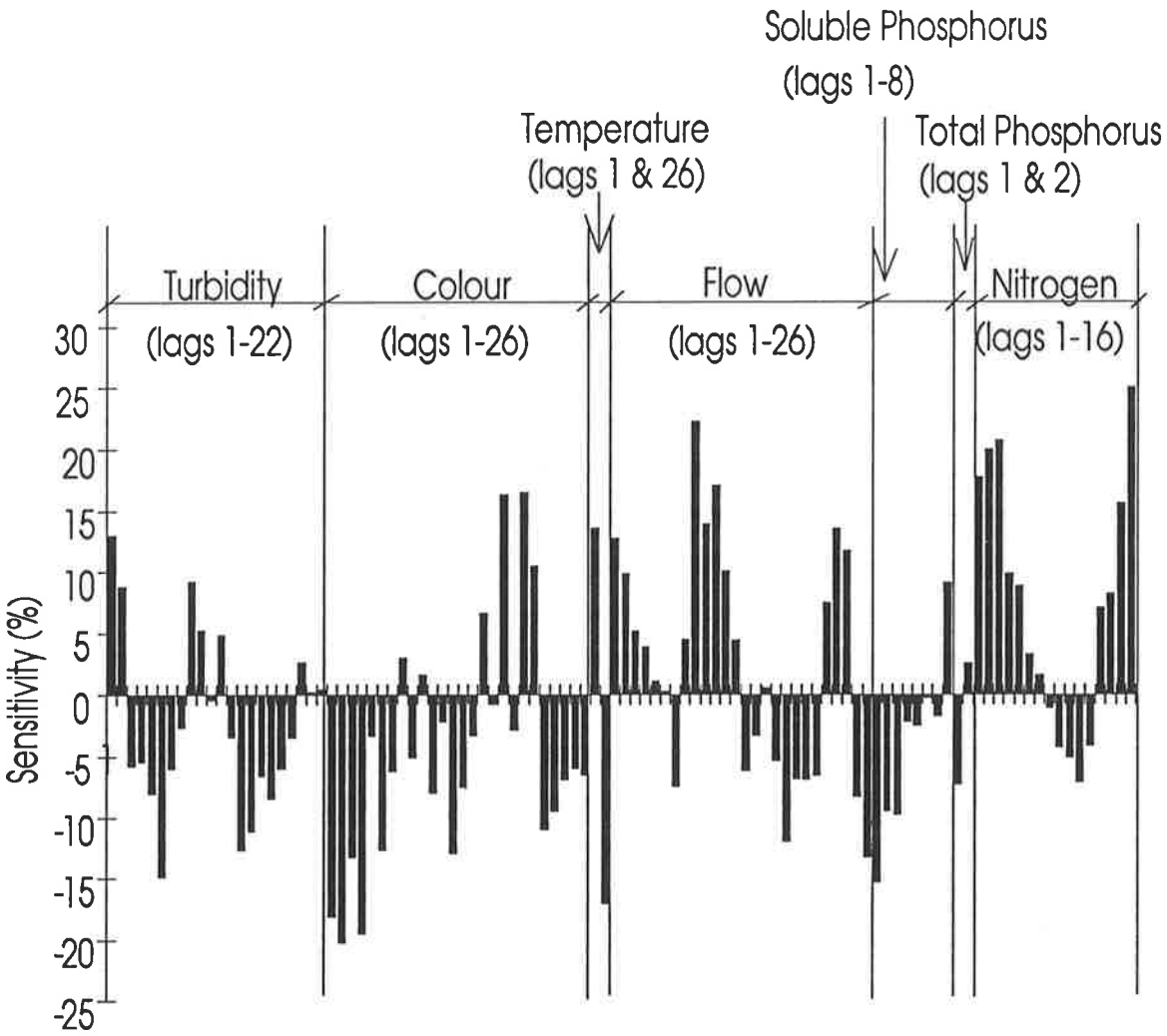


Figure 4.30: Typical Plot of the Relative Significance of the Model Inputs for the Model with 102 Inputs

The sensitivity analysis indicates that the size of the algal population is inversely related to total and soluble phosphorus. However, as explained in Section 4.3.1.1, this is merely an indication that phosphorus is not limiting, and that high levels of phosphorus are strongly correlated with high turbidity values. The model shows a positive correlation between the concentration of *Anabaena* and nitrogen, suggesting that nitrogen is the limiting nutrient. This is surprising, as *Anabaena* have the ability to fix nitrogen from the atmosphere.

4.5 Conclusions

The results obtained in this chapter indicate that multivariate ANN models are suitable for forecasting the relative magnitude and timing of incidences of *Anabaena* in the River Murray at Morgan up to 4 weeks in advance.

The causal variables were found to interact in a very complex manner, as indicated by the fact that model performance increased with an increase in the number of inputs (i.e. the model with 102 inputs performed better than the model with 43 inputs, which performed better than the model with 7 inputs), and by the results of the sensitivity analysis carried out on the model with 102 inputs. No one variable was found to be dominant, with all input variables contributing. There appears to be a strong inverse relationship between *Anabaena* concentrations and colour and turbidity, and a strong positive relationship between *Anabaena* concentrations and temperature. Nitrogen, rather than phosphorus, seems to be the limiting nutrient, which is surprising, as *Anabaena* have the ability to fix nitrogen from the atmosphere. In recent years, algal blooms appear to occur on the falling limb of a flood hydrograph, suggesting that algae might be advected from anabranches or lagoons into the main channel of the river.

The results obtained indicate that there is a definite advantage in considering lagged inputs. One would suspect that this is especially true when changes in the input variables with time are rapid. Consequently, using lagged inputs is of greater advantage when forecasting blue-green algae in rivers, rather than lakes, as river environments are dominated by the flow regime.

The fact that the relative significance and timing of most significant incidences of *Anabaena* were successfully predicted by the model developed indicates that the environmental variables used as model inputs (i.e. colour, turbidity, temperature, flow, total phosphorus, soluble phosphorus and nitrogen) are the primary causal factors of incidences of *Anabaena* in the River Murray at Morgan. However, the forecasts could possibly be improved by including other causal variables such as the amount of incoming radiation, the concentrations of algal predators and competing algal species, the thermal structure of the river, pH and dissolved oxygen, for which no data were available in the case study considered.

Chapter 5

Conclusions and Recommendations

5.1 Contributions of the Research

This thesis has made the following contributions:

1. The suitability of back-propagation ANNs for modelling multivariate water quality time series has been assessed by applying them to two diverse case studies, including:

   (i) The long-term forecasting of salinity in the River Murray at Murray Bridge, South Australia, given past values of salinity, flow and river levels at Murray Bridge and various upstream locations.

   (ii) The prediction of the incidence of blue-green algae (*Anabaena spp.*) in the River Murray at Morgan, South Australia, given various prevailing environmental conditions, including the colour, turbidity, temperature and flow of the river and the concentrations of total phosphorus, soluble phosphorus and total nitrogen.

2. The performance of back-propagation ANN models (univariate and multivariate) has been compared with that of ARMA type models (univariate and multivariate), which are commonly used for time series modelling. Both model types (i.e. ANN and ARMA) were applied to the case study of forecasting salinity in the River Murray at Murray Bridge, and compared in terms of generalisation ability and ease of use.

3. Guidelines for the use of back-propagation ANNs for modelling multivariate time series have been developed. In particular, information is given on:

   - The ability of multilayer back-propagation networks to deal with non-stationary time series.
   - How to best use the available training data.
   - How to determine appropriate model inputs.
   - The best way to obtain multi-step forecasts.
   - The effect of using different internal network parameters, including the learning rate, the momentum, the epoch size, the transfer function, the error function and initial weight distribution.
   - The effect of using the delta-bar-delta algorithm.
   - The way the generalisation ability of ANN models changes as training progresses, especially once a local minimum in the error surface has been reached.
   - Methods for optimising network performance.
   - How to best perform real time forecasting simulations.
   - The effect of using different network geometries, including the effect of the number of hidden layers and the number of nodes per hidden layer.

4. The method of Haugh and Box (1977), which is commonly used to determine the inputs for multivariate time series models of the ARMA type, has been suggested as a viable means of determining the inputs for multivariate ANN models. In addition, a new method for determining the inputs for multivariate ANN models has been introduced, which is based on a neural network approach. The performance of both methods was assessed by applying them to the case study of forecasting salinity in the River Murray at Murray Bridge. The neural network based method was also applied to the case study of modelling the incidence of blue-green algae in the River Murray at Morgan.

5. It has been demonstrated that back-propagation ANNs can be used to obtain valuable information about the relationships between environmental time series by carrying out sensitivity analyses, and do not have to be treated as a "black box".

6. It has been shown that back-propagation ANNs can be successfully used to address some of the water quality issues in the River Murray. They are capable of producing useful 28-day forecasts of salinity at Murray Bridge, which can be used to reduce the salinity of the water supplied to users in Adelaide. In addition, they can successfully forecast the timing and relative magnitude of incidence of *Anabaena* at Morgan up 4 weeks in advance, and provide useful information about the causal factors of blue-green algal blooms.

5.2 General Conclusions

1. Back-propagation ANNs are a suitable tool for modelling multivariate water quality time series. The following properties make them particularly useful for the above task:

   - They perform well when limited data are available.
   - They are able to model non-stationary time series, thus reducing the amount of data pre-processing (and post-processing) required.
   - They are suitable for longer-term forecasting, as they can be trained for specific forecasting periods.
   - They are capable of determining complex relationships between time series.
   - They are non-linear.
   - They are relatively insensitive to model and data noise.
   - They are capable of providing useful information about the relationships between water quality time series.

2. Multivariate ANN models have the following advantages over multivariate ARIMA models:

   - They perform better when longer-term forecasts are required.
   - They perform better when limited data are available.
   - They are non-linear.
   - They do no require stationary input data.
   - They can be trained for a specific forecasting period.
   - They are less sensitive to model noise.

   The disadvantages of multivariate ANN models, compared with multivariate ARIMA models, include:

   - They give poorer short-term forecasts.
   - There are some difficulties in optimising model performance (i.e. obtaining the best estimates of the connection weights).
   - Optimum network geometries and internal parameters generally have to be obtained using a trial and error approach.

3. Using analytical procedures for determining the inputs for multivariate ANN models has a number of advantages, such as reducing the number of inputs and consequently network geometry, training time and the chance of the network getting stuck in undesirable regions of the weight space. Such methods are especially useful when no *a priori* knowledge is available to suggest possible model inputs, and for complex problems, where the number of potential inputs is large. The following two methods were found to be appropriate for determining the inputs for multivariate ANN models:

   - The method of Haugh and Box (1977), which is commonly used to determine the inputs for multivariate time series models of the ARMA type.

   - A neural network based method, which involves the development of simple univariate and bivariate ANN models to establish relationships between the output time series and each of the input time series. The strength of these relationships is determined with the aid of sensitivity analyses, and the inputs that have a significant effect on the outputs are chosen as inputs to the multivariate model.

The latter method has the following advantages:

- It can determine which inputs are significant for a specific forecasting period, potentially reducing the number of inputs.

- It is simpler and quicker to use, as data pre-processing (e.g. differencing, pre-whitening) is not required.

One disadvantage of the neural network based method is that there is some judgement involved in determining the level of significance, above which inputs of the univariate and bivariate models are included in the multivariate model.

4. The following guidelines / information should be taken into account when modelling multivariate time series using back-propagation ANNs:

   - ANNs are able to cater for non-stationarities in the data (i.e. trends, seasonal variations) with the aid of their hidden layer nodes.

   - In order to adequately assess the generalisation ability of ANN models, it is desirable to use as much data as possible for validation purposes. A good way to achieve this is to train a number of models, each using different data for training and testing. In this way, independent forecasts can be obtained for all available data, while maximising the amount of data used for training.

   - The neural network based method described in 3 above should be used to determine appropriate model inputs.

   - In cases where multi-step forecasts are required, the models should be trained for the required forecasting period(s) directly. Each node in the output layer should correspond to a forecast with a particular forecasting period.

   - The size of the steps taken in weight space, and hence training speed, are significantly affected by the learning rate, momentum, epoch size (depending on the learning rate used), error function and gain of the transfer function. It should be noted that the same learning speed can be achieved by using different combinations of the above parameters.

- The learning rate, momentum, epoch size and initial weight distribution do not have a significant effect on the generalisation ability of ANNs. However, if the combination of the above parameters is such that the steps taken in weight space are too large, divergent behaviour may occur during training.

- The transfer and error functions can have a significant effect on network performance. Consequently, it is suggested to assess the effect of different transfer and error functions in order to optimise model performance.

- There does not appear to be any advantage in using the delta-bar-delta algorithm (Jacobs, 1988), as a trial and error approach has to be used to optimise certain parameters, which control the rate at which the learning rates are adjusted during training.

- During training, the generalisation ability of ANNs increases until a local minimum in the error surface is reached. The learn count at which this occurs is a function of the size of the steps taken in weight space. Once a local minimum in the error surface has been reached, the generalisation ability of the network increases and decreases in an oscillatory fashion with continued training, as the network jumps from one side of a local minimum to the other or from one local minimum to another. These oscillations generally take place after each weight update, unless very small steps are taken in weight space. The magnitude of the oscillations is a function of the size of the steps taken in weight space. Greater steps in weight space enable networks to escape local minima in the error surface, allowing better solutions to be found. However, if the steps taken in weight space are too large, divergent behaviour might occur. The network behaviour described above has the following implications for users of back-propagation networks:

  (i) If a fixed number of training samples are presented to networks, their generalisation ability cannot be monitored as training progresses. Consequently, no information is available on the learn count at which a local minimum in the error surface is reached, and the magnitude of the oscillations in generalisation ability once a local minimum in the error surface has been reached. However, without this information, it is very difficult to obtain good generalisation ability, as illustrated by the following examples:
     - Training might be stopped before a local minimum in the error surface has been reached.

- If large steps are taken in weight space, the oscillations in the generalisation ability of the network, once it has reached a local minimum in the error surface, will be large. Consequently, even if training is stopped once a local minimum in the error surface has been reached, there is a strong likelihood that, at the learn count when training is stopped, the generalisation ability of the network is far from optimal.
- If small steps are taken in weight space and a local minimum in the error surface has been reached, the generalisation ability of the network is almost independent of when training is stopped. However, as small steps are taken in weight space, the network is less likely to escape local minima in the error surface, resulting in inferior generalisation ability. In addition, training time is increased.

The above information also has to be taken into account when optimising the internal network parameters using a trial and error approach. As discussed above, the combination of internal network parameters affects the size of the steps taken in weight space. Consequently, the difference in generalisation ability of two networks with different internal parameters at a particular learn count does not necessarily reflect the generalisation capabilities of the two networks. In other words, the comparative generalisation ability of two networks with different internal parameters might be different at different learn counts.

(ii) Cross-validation should be used as the stopping criterion, as it enables the generalisation ability of the network to be monitored as training progresses. This enables larger steps to be taken in weight space, thus increasing training speed and the ability of the networks to escape local minima in the error surface. The following procedure can be used to obtain near-optimum network performance:

- The network should be trained using a small epoch size, a relatively large learning rate and a relatively large momentum. It should be noted that the absolute values of learning rate and momentum that should be used are dependent on the nature of the error surface. The learning rate and momentum chosen should be small enough to avoid divergent behaviour.
- Training should be stopped at regular intervals, and the generalisation ability of the network assessed. It should be noted that smaller testing intervals increase the chances of obtaining networks with better

generalisation ability. It is advisable to reduce the testing intervals once a network approaches a local minimum in the error surface.

- Training should be stopped once a local minimum in the error surface has been reached.
- The network that produced the best forecast should be selected and training re-commenced after reducing the learning rate and momentum to very small values (possibly reducing them by a factor of 100) and setting the epoch size equal to the number of samples in the training set.
- Training should be stopped, and the generalisation ability of the network assessed, after each weight update (i.e. the testing interval should be equal to the epoch size).
- Training should be continued until there is no further improvement in the network's generalisation ability.

- When using cross-validation as the stopping criterion for real time forecasting simulations, it is necessary to have separate training, testing and forecasting sets. The network that gives the best forecasts for the testing set should be used for forecasting. In order to obtain good generalisation ability, and hence good real time forecasts, the testing set has to be chosen carefully, so that it is representative of the relationship to be approximated.

  When limited data are available, it might be better to present a fixed number of training samples to the network during training, in order to utilise all the available data in the training phase. However, it is vital to carry out some exploratory analysis, in which a small portion of the training set is used to test the performance of the network at various stages of learning. This gives an indication of how many training samples need to be presented to the network until a local minimum in the error surface is reached and the magnitude of the oscillations in generalisation ability, once a local minimum in the error surface has been reached, for a particular combination of network parameters. The knowledge obtained as part of the exploratory phase (i.e. what network parameters are appropriate and how many training samples should be presented to the network) can then be used to re-train the network, using all available data for training. The knowledge of when a local minimum in the error surface is reached may also be used to reduce the size of the steps taken in weight space at that point, in order to reduce the magnitude of the oscillations in generalisation ability.

- The number of hidden layers (i.e. one or two) and the number of nodes per hidden layer do not appear to have a significant effect on generalisation ability. However, the effect of network geometry is problem dependent. Training speed is significantly affected by network geometry, as the time taken to process one training sample is a function of the number of connection weights.

- Once training is complete, sensitivity analyses can be used to obtain useful information about the relationships between the network inputs and outputs

5.3 Specific Conclusions

5.3.1 Case Study 1 - Forecasting Salinity in the River Murray

1. Back-propagation ANNs are a suitable tool for obtaining long-term salinity forecasts in the River Murray at Murray Bridge. The RMSE and AAPE of 14 day forecasts obtained as part of a real time forecasting simulation for 1991 were 48.1 EC units and 6.4% respectively. The corresponding values were 78.3 EC units and 9.8% for the 28 day forecast.

2. Multivariate ANN models are better suited to longer-term forecasting of salinity in rivers than multivariate ARIMA models.

3. Multivariate ARIMA models are better suited to short-term (several steps ahead) forecasting of salinity in rivers than multivariate ANN models.

4. Using upstream salinities and flow information does not significantly improve short-term salinity forecasts. However, upstream salinities and flow become increasingly important as the forecasting period increases. The inclusion of river level inputs is not necessary.

5.3.2 Case Study 2 - Modelling Blue-Green Algae in the River Murray

1. Back-propagation ANNs are a useful tool for forecasting the incidence of a species group of the cyanobacterium *Anabaena spp.* in the River Murray at Morgan up to 4 weeks in advance. Over a test period of eight years, the model was able to correctly forecast whether the incidence of *Anabaena* was significant (i.e. $\geq$ 1000 cells/mL) in

any particular year. In addition, the timing of four of the five significant events was predicted correctly.

2. The environmental variables that affect the size of the population of *Anabaena* interact in a very complex manner. No one variable was found to be dominant, with all variables contributing approximately equally. In addition, different environmental conditions appear to be responsible for different significant incidences of *Anabaena*, making it difficult to put appropriate management practices in place to prevent large incidences of *Anabaena* from occurring. There appears to be a strong inverse relationship between *Anabaena* concentrations and colour and turbidity, and a strong positive relationship between *Anabaena* concentrations and temperature. Nitrogen, rather than phosphorus, seems to be the limiting nutrient. This is surprising, as *Anabaena* are able to fix nitrogen from the atmosphere. In recent years, algal blooms appear to occur on the falling limb of a flood hydrograph, suggesting that algae might be advected from anabranches or lagoons into the main channel of the river.

3. There is a definite advantage in using lagged inputs when modelling the incidence of blue-green algae in rivers.

5.4 Recommendations for Further Work

1. The results of this research indicate that there are some difficulties in optimising feedforward networks using the back-propagation learning algorithm, especially in real time forecasting situations. Consequently, it might be worthwhile to compare the performance of the back-propagation algorithm with random and statistical approaches for optimising the connection weights, such as the Boltzmann machine (Ackley et al., 1985), random directed search networks (Matyas, 1965; Solis and Wets, 1985) and genetic algorithms (Goldberg, 1989).

2. Although this research suggests that network geometry does not have a significant effect on generalisation ability, it might be worthwhile to investigate the effect of using techniques for adapting network geometry as training progresses, such as those suggested by Fahlman and Lebiere (1990), Weigend et al. (1990), Sietsma and Dow (1991) and Chung and Lee (1992).

3. For the case studies considered in this research, the performance of the ANN models should be compared with other types of models, such as physically based and TVP models (Young, 1994).

4. The guidelines for the development of multivariate ANN models suggested in this research should be applied to other case studies to investigate whether they may be considered general or whether some of them are specific to the case studies considered.

Appendix A

Notation

a	constant
b	constant
B	backshift operator
c	constant
d	degree of non-seasonal differencing
d'	number of observations lost as part of differencing
d^{DW}	Durbin-Watson statistic
D	degree of seasonal differencing
e_t	random noise component of a time series at discrete time t
$\mathbf{e}_t$	N' x 1 vector of random noise components at discrete time t
$\mathbf{E}_t$	vector of random noise components for a VARIMA model in state space form at time t
$E^G(t)$	global error function at discrete time t (ANN)
E^L	local error at a particular processing element (ANN)
E^N	energy function (ANN)
$f^{rb}{}_j(.)$	radial basis function at node j (ANN)
$\mathbf{f}_t$	g' x 1 vector of random noise components at discrete time t
$F(.)$	transfer function (ANN)
FE	RMS forecasting error (ANN)

FE_{ave}	average RMS forecasting error (ANN)
$FE_{plat,ave}$	average RMS forecasting error, once a local minimum in the error surface has been reached (ANN)
$FE_{plat,max}$	maximum RMS forecasting error, once a local minimum in the error surface has been reached (ANN)
$FE_{plat,min}$	minimum RMS forecasting error, once a local minimum in the error surface has been reached (ANN)
g	forecasting period
$\mathbf{H}_t$	covariance matrix of $\mathbf{e}_t$ at discrete time t
i	network input (ANN)
i	vector containing the network inputs (ANN)
I_j	activation level of node j (ANN)
$\mathbf{j}_t$	optimal estimator of the state vector
$\mathbf{I}_k$	identity matrix
k'	number of autocorrelations used in the calculation of the Box-Pierce chi-square statistic (Q^{BP})
k_{AR}	number of AR parameters at small (i.e. non-seasonal) lags used to describe time series i
k_{MA}	number of MA parameters at small (i.e. non-seasonal) lags used to describe time series i
k_{max}	maximum lag of the inputs (excluding any inputs at seasonal lags) to univariate or bivariate ANN models
K	maximum lag for which the sample autocorrelation or sample partial autocorrelation should be calculated to determine whether a time series is stationary
K'	number of exogenous variables
L	number of seasons in one year
L(.)	likelihood function
M_i	order statistic median of an observation for order i
$\overline{M}$	sample mean of the order statistic median values
m	number of values in the state vector (φ_t)
m'	number of degrees of freedom
n	number of observations in the time series
n'	year index
n_p	total number of model parameters
$\mathbf{n}_t$	N'x1 vector of prediction errors at discrete time t
n_y	number of years of data available
N	variable

N'	number of time series forming part of a multivariate time series model
N^{H1}	number of nodes in the first hidden layer (ANN)
N^{H2}	number of nodes in the second hidden layer (ANN)
N^{I}	number of nodes in the input layer (ANN)
N^{K}	number of nodes in the Kohonen layer of an LVQ network (ANN)
N^{O}	number of nodes in the output layer (ANN)
N^{p}	number of nodes in the previous layer (ANN)
N^{par}	total number of parameters (AR and MA) in a VARIMA model
N^{TOT}	total number of nodes in the entire network (ANN)
N^{w}	number of connection weights (ANN)
NAR_{ij}	maximum lag of the AR parameters from time series j used to describe time series i
o'	forecasting origin
$o_k^d(t)$	desired (actual, historical) output at discrete time t (ANN)
$o_k^p(t)$	predicted output at discrete time t (ANN)
$\mathbf{o}^P$	vector containing the predicted network outputs (ANN)
p	order of the non-seasonal autoregressive parameters
P	order of the seasonal autoregressive parameters
p^t	period of the trend component
$\mathbf{P}_{t-1}$	m x m covariance matrix of the estimation error
P^{TE}	number of patterns (samples) in the testing set (ANN)
P^{TR}	number of patterns (samples) in the training set (ANN)
q	order of the non-seasonal moving average parameters
q^c	lag at which the theoretical ACF and PACF cut off
Q	order of the seasonal moving average parameters
$\mathbf{Q}_t$	covariance matrix of $\mathbf{f}_t$
Q^{BP}	Box-Pierce chi-square statistic
r'	order of the parameters of the exogenous variables
r_k	sample autocorrelation at lag k
r_{kk}	sample partial autocorrelation at lag k
$r_{z_i z_j}(k)$	sample cross-correlation function between time series $\{z_{i,t}\}$ and $\{z_{j,t}\}$
$\hat{r}$	PPCC test statistic
R_t	residual at discrete time t
s_t	seasonal component at discrete time t
$s_{z_i}(k)$	sample autocovariance function at for time series $\{z_{i,t}\}$ at lag k
$s_{z_i z_j}(k)$	sample cross-covariance function between time series $\{z_{i,t}\}$ and $\{z_{j,t}\}$
T'_j	threshold for node j (ANN)
$\mathbf{T}_t$	m x m transition matrix in the transition equation at discrete time t

u_t	observation at discrete time t (transformed data)
$\{u_t\}$	time series of observations (transformed data)
$\mathbf{v}_t$	K' x 1 vector of exogenous variables at discrete time t
w_{ji}	connection weight between nodes i and j (ANN)
W^N	matrix containing the network connection weights (ANN)
x_i	input from node i (ANN)
y_j	output of node j (ANN)
$\mathbf{Y}_t$	N' x m transition matrix in the measurement equation at discrete time t
z_t	observation at discrete time t (raw data)
$\mathbf{z}_t$	N' x 1 vector of observations at discrete time t
$\mathbf{Z}_t$	state vector for a VARIMA model in state space form at time t
$\{z_t\}$	time series of observations (raw data)
$\bar{z}$	sample mean of the time series $\{z_t\}$
$\bar{z}_t$	trend component at discrete time t (mean function)
α	parameter
β	parameter
β^v	vigilance parameter (ANN)
$\chi^2_5(k'-n_p)$	point on the chi-square distribution having $(k'-n_p)$ degrees of freedom such that there is an area of 0.05 under the curve of this distribution above this point
Δw_{ji}	amount the connection weight between nodes i and j is changed during one weight update (ANN)
ε	epoch size (ANN)
ϕ_i	autoregressive parameter
Φ_i	N' x N' matrix of AR parameters
$\Phi_n(z)$	cumulative distribution function of the standard normal distribution
Γ_k	N' x K' matrix of parameters for the exogenous variables
η	learning rate (ANN)
φ_t	state vector at discrete time t
λ	coefficient
μ	momentum value (ANN)
$\mu^{th}(t)$	theoretical mean at discrete time t
ν	seasonal index
Π_t	covariance matrix of $\mathbf{n}_t$ at discrete time t
θ_i	moving average parameter
Θ_j	N' x N' matrix of MA parameters
ρ_k	theoretical autocorrelation at lag k
ρ_{kk}	theoretical partial autocorrelation at lag k
$\rho_{z_i z_j}(k)$	theoretical cross-correlation function between time series $\{z_{i,t}\}$ and $\{z_{j,t}\}$

$\sigma_{z_i}(k)$	theoretical autocovariance function at for time series $\{z_{i,t}\}$ at lag k
$\sigma_{z_i z_j}(k)$	theoretical cross-covariance function between time series $\{z_{i,t}\}$ and $\{z_{j,t}\}$
$\sigma^2(t)$	variance at discrete time t
τ	width of a time strip
ξ	parameter
∇	first difference operator

Appendix B

Abbreviations

AAE	Average absolute error
AAPE	Average absolute percentage error
ACF	Autocorrelation function
ADALINE	Adaptive linear element
AIC	Akaike's information criterion
ANN	Artificial neural network
AR	Autoregressive
ARIMA	Integrated mixed autoregressive - moving average
ARMA	Autoregressive - moving average
ARMAX	Exogenous-variable autoregressive - moving average
ART	Adaptive resonance theory
BAM	Bi-directional associative memory
BIC	Bayes information criterion
CARMA	Contemporaneous autoregressive - moving average
CAT	Criterion autoregressive transfer function
CCF	Cross-correlation function
CDE	Cumulative delta (learning rule)
CPU	Central processing unit
dbd	Delta-bar-delta

DE	Delta (learning rule)
DHR	Dynamic harmonic regression
edbd	Extended delta-bar-delta
EM	Expectation step - maximisation step
FL1L	Flow at Lock 1 Lower
FL7L	Flow at Lock 7 Lower
FOC	Flow at Overland Corner
GRW	Generalised random walk
GTF	General transfer function
IPRM	Intelligent pattern recognition memory
IRW	Integrated random walk
LL1L	River level at Lock 1 Lower
LL1U	River level at Lock 1 Upper
LLO	River level at Loxton
LMB	River level at Murray Bridge
LMN	River level at Mannum
LMO	River level at Morgan
LOC	River level at Overland Corner
LPS	Lipopolysaccharide
LWE	River level at Waikerie
LVQ	Learning vector quantisation
MA	Moving average
MADALINE	Multiple adaptive linear element
MAE	Maximum absolute error
MANN	Multivariate artificial neural network
MIB	2-Methylisoborneol
MMSE	Maximum mean squared estimator
M-P	McCulloch - Pitts
MSE	Median squared error
NCDE	Normalised cumulative delta (learning rule)
NEP	Network emulation processor
NNN	Natural neural network
PACF	Partial autocorrelation function
PARMA	Periodic autoregressive - moving average
PLC	Primary liver cancer
PPCC	Probability plot correlation coefficient
PSPs	Paralytic shellfish poisons
RMSE	Root mean squared error
SL5U	Salinity at Lock 5 Upper

SLO	Salinity at Loxton
SMB	Salinity at Murray Bridge
SMN	Salinity at Mannum
SMO	Salinity at Morgan
SWE	Salinity at Waikerie
SOM	Self-organising map
SSD	Sequential spectral decomposition
TFN	Transfer function autoregressive - moving average
TI	Testing interval
TVP	Time variable parameter
UANN	Univariate artificial neural network
VARIMA	Vector autoregressive integrated moving average
VARMA	Vector autoregressive - moving average
VLSI	Very large scale integration

Appendix C

Training / Testing Sets

C.1 Forecasting Salinity in the River Murray

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
01	SMB	1, 2, 3, 364, 365, 366, 367	0
02	SMB	1, 365, 366	0
03	SMB	1, 2, 3, ..., 19, 20, 354, 355, ..., 375, 376	0, -7, -13
04	SMB	1, 2, 3, ..., 14, 15	0, -7, -13
05	SMB	1, 2, 3, ..., 19, 20, 354, 355, ..., 375, 376	-13
06	SMB	1, 2, 3, ..., 14, 15	0, -4, -13

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
07	SMB	1, 2, 3, ..., 16	-13
	SMN	1, 2, 3, ..., 33	
	SMO	1, 3, 5, ..., 31	
	SWE	1, 2, 3, ..., 11	
	SLO	1, 2, 3, ..., 11	
	FL1L	-23, -22, ..., -11	
		-9, -4, ..., 41	
	FOC	-9, -4, ..., 21	
	LMB	1, 6, 11	
	LMN	1, 6, 11	
	LL1U	1, 6, 11	
	LL1L	1, 6, 11	
	LMO	1, 6, 11	
	LWE	1, 6, 11	
	LOC	-10, -14, -9	
	LLO	1, 6, 11	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
08	SMB	1, 3, 5, ..., 9	-13
	SMN	1, 3, 5, ..., 15	
	SMO	1, 2, 3	
	SWE	1, 2, 3, ..., 7	
	SLO	1, 2, 3, ..., 6	
	FL1L	-23, -22, ..., -20	
		-9, -4, ..., -6 and 41	
	FOC	-9, -4, ..., 21	
	LMB	1, 6, 11	
	LMN	1, 6, 11	
	LL1U	1, 6, 11	
	LL1L	1, 6, 11	
	LMO	1, 6, 11	
	LWE	1, 6, 11	
	LOC	-10, -14, -9	
	LLO	1, 6, 11	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
09	SMB	1, 2, 3, 4	-13
	SMN	1, 3, 5, ..., 15	
	SMO	1, 2, 3	
	SWE	1, 2, 3	
	SLO	1, 2, 3, 5	
	FOC	-15, -13, ..., -1	
	LMB	1, 3, ..., 15	
	LL1U	6, 11, 16	
	LL1L	6, 11, 16	
	LOC	-14, -12, ..., -4	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
10	SMB	1, 3, 5, ..., 11	-13
	SMN	1, 3, 5, ..., 15	
	SMO	1, 3, 5, ..., 15	
	SWE	1, 2, 3, 5	
	SLO	1, 2, 3, 5	
	FOC	-19, -17, ..., 7	
	LL1U	-3, -1, ..., 5	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
11	SMB	1, 3, 5, ..., 11	-1, -3, ..., -13
	SMN	1, 3, 5, ..., 15	
	SMO	1, 3, 5, ..., 15	
	SWE	1, 2, 3, 5	
	SLO	1, 2, 3, 5	
	FOC	-19, -17, ..., 7	
	LL1U	-3, -1, ..., 5	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
12	SMB	1, 3, 5, ..., 11	0, -4, -13
	SMN	1, 3, 5, ..., 15	
	SMO	1, 3, 5, ..., 15	
	SWE	1, 2, 3, 5	
	SLO	1, 2, 3, 5	
	FOC	-19, -17, ..., 7	
	LL1U	-3, -1, ..., 5	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
13	SMB	1, 3, 5, ..., 11	0, -4, -13
	SMN	1, 3, 5, ..., 15	
	SMO	1, 3, 5, ..., 15	
	SWE	1, 2, 3, 5	
	SLO	1, 2, 3, 5	
	FOC	1, 3, 5, 7	
	LL1U	1,3, 5	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
14	SMB	1, 3, 5, ..., 11	-27
	SMN	1, 3, 5, ..., 15	
	SMO	1, 3, 5, ..., 15	
	SWE	1, 2, 3, 5	
	SLO	1, 2, 3, 5	
	SL5U	1, 2, 3, ..., 10	
	FOC	-19, -17, ..., 7	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
15	SMB	1, 3, 5, ..., 11	-27
	SMN	1, 3, 5, ..., 15	
	SMO	1, 3, 5, ..., 15	
	SWE	1, 2, 3, 5	
	SLO	1, 2, 3, 5	
	SL5U	1, 2, 3, ..., 10	
	FL7L	-13, -11, ..., 13	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
16	SMB	1, 3, 5, ..., 11	-27
	SMN	1, 3, 5, ..., 15	
	SMO	1, 3, 5, ..., 15	
	SWE	1, 2, 3, 5	
	SLO	1, 2, 3, 5	
	SL5U	1, 2, 3, ..., 10	
	FL7L	1, 3, ..., 27	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
17	SMB	1, 2, ..., 20, 354, 355, ..., 376	-13

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
18	SMB		-13
	SMN	1, 3, ..., 30	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
19	SMB		-13
	SMO	1, 3, ..., 30	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
20	SMB SWE	 1, 3, ..., 30	-13

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
21	SMB SLO	 1, 3, ..., 30	-13

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
22	SMB FL1L	 1, 3, ..., 30	-13

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
23	SMB FL1L	 1, 3, ..., 50	-13

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
24	SMB SMN SMO SWE SLO FL1L	1, 2 1, 2, 3, 4 1, 2, 3, 4, 5, 6, 7 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	-13

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
25	SMB SMN SMO SWE SLO FL1L	1, 2 1, 2 1, 2 1, 2, 3, 4 1, 2, 3, 4, 5, 6, 7 1, 2, 3, 4, 5, 6, 7, 8	-13

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
26	SMB	1, 3, 5, ..., 11	-13
	SMN	1, 3, 5, ..., 15	
	SMO	1, 3, 5, ..., 15	
	SWE	1, 2, 3, 5	
	SLO	1, 2, 3, 5	
	FL1L	1, 3, 5, 7	

C.2 Modelling Blue-Green Algae in the River Murray

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
01	*Anabaena* Colour	1, 2, ..., 26	-1

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
02	*Anabaena* Turbidity	1, 2, ..., 26	-1

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
03	*Anabaena* Temperature	1, 2, ..., 26	-1

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
04	*Anabaena* Flow	1, 2, ..., 26	-1

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
05	*Anabaena* Soluble Phosphorus	1, 2, ..., 26	-1

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
06	*Anabaena* Total Phosphorus	1, 2, ..., 26	-1

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
07	*Anabaena* Nitrogen	1, 2, ..., 26	-1

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
08	*Anabaena*		-1, -3
	Colour	1	
	Turbidity	1	
	Temperature	1	
	Flow	1	
	Soluble Phosphorus	1	
	Total Phosphorus	1	
	Nitrogen	1	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
09	*Anabaena*		-1, -3
	Colour	1, 2, 3, 4	
	Turbidity	1, 2, ..., 8	
	Temperature	1, 26	
	Flow	1, 2, ..., 15	
	Soluble Phosphorus	1, 2, ..., 5	
	Total Phosphorus	1, 2	
	Nitrogen	1, 2, ..., 7	

Training / Testing Set	Variable	Lags of Inputs	Lags of Outputs
10	*Anabaena*		-1, -3
	Colour	1, 2, ..., 26	
	Turbidity	1, 2, ..., 22	
	Temperature	1, 26	
	Flow	1, 2, ..., 26	
	Soluble Phosphorus	1, 2, ..., 8	
	Total Phosphorus	1, 2	
	Nitrogen	1, 2, ..., 16	

Appendix D

Neural Network Parameters and Geometries

D.1 Forecasting Salinity in the River Murray

Parameter / Geometry	Model Number				
	01	02	03	04	05
Learning Rate (η)	0.001	0.005	0.02	0.02	0.02
Momentum (μ)	0.6	0.6	0.6	0.6	0.6
Epoch Size (ε)	5	5	5	5	5
Transfer Function	Linear	Linear	Linear	Linear	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	7	7	7	3	3
No. of Nodes in Hidden Layer 1 (N^{H1})	0	0	0	0	0
No. of Nodes in Hidden Layer 2 (N^{H2})	0	0	0	0	0
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	06	07	08	09	10
Learning Rate (η)	0.02	0.02	0.02	0.02	0.02
Momentum (μ)	0.6	0.6	0.6	0.6	0.6
Epoch Size (ε)	5	5	5	5	5
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	3	3	3	7	41
No. of Nodes in Hidden Layer 1 (N^{H1})	6	3	6	15	0
No. of Nodes in Hidden Layer 2 (N^{H2})	0	0	2	5	0
No. of Outputs (N^O)	1	1	1	1	3

Parameter / Geometry	Model Number				
	11	12	13	14	15
Learning Rate (η)	0.1	0.1	0.1	0.1	0.05
Momentum (μ)	0.6	0.6	0.6	0.6	0.6
Epoch Size (ε)	5	5	5	5	5
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	41	41	15	41	15
No. of Nodes in Hidden Layer 1 (N^{H1})	45	60	24	45	24
No. of Nodes in Hidden Layer 2 (N^{H2})	15	20	8	15	8
No. of Outputs (N^O)	3	3	3	1	3

Parameter / Geometry	Model Number				
	16	17	18	19	20
Learning Rate (η)	Default	Default	Default	Default	0.005
Momentum (μ)	Default	Default	Default	Default	0.6
Epoch Size (ε)	16	16	16	16	5
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	141	69	51	51	51
No. of Nodes in Hidden Layer 1 (N^{H1})	90	45	45	45	45
No. of Nodes in Hidden Layer 2 (N^{H2})	30	15	15	15	15
No. of Outputs (N^O)	1	1	1	7	1

Parameter / Geometry	Model Number				
	21	22	23	24	25
Learning Rate (η)	0.005	0.005	0.005	0.005	0.005
Momentum (μ)	0.6	0.6	0.6	0.6	0.6
Epoch Size (ε)	10	16	35	85	180
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	51	51	51	51	51
No. of Nodes in Hidden Layer 1 (N^{H1})	45	45	45	45	45
No. of Nodes in Hidden Layer 2 (N^{H2})	15	15	15	15	15
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	26	27	28	29	30
Learning Rate (η)	0.005	0.005	0.1	0.1	0.1
Momentum (μ)	0.6	0.6	0.6	0.6	0.6
Epoch Size (ε)	365	1671	5	10	16
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	51	51	51	51	51
No. of Nodes in Hidden Layer 1 (N^{H1})	45	45	45	45	45
No. of Nodes in Hidden Layer 2 (N^{H2})	15	15	15	15	15
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	31	32	33	34	35
Learning Rate (η)	0.1	0.005	0.005	0.005	0.005
Momentum (μ)	0.6	0.05	0.05	0.05	0.4
Epoch Size (ε)	35	5	365	16	16
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	51	51	51	51	51
No. of Nodes in Hidden Layer 1 (N^{H1})	45	45	45	45	45
No. of Nodes in Hidden Layer 2 (N^{H2})	15	15	15	15	15
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	36	37	38	39	40
Learning Rate (η)	0.005	0.005	0.02	0.1	0.2
Momentum (μ)	0.8	0.9	0.05	0.05	0.05
Epoch Size (ε)	16	16	16	16	16
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	51	51	51	51	51
No. of Nodes in Hidden Layer 1 (N^{H1})	45	45	45	45	45
No. of Nodes in Hidden Layer 2 (N^{H2})	15	15	15	15	15
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	41	42	43	44	45
Learning Rate (η)	0.02	0.1	0.2	0.02	0.2
Momentum (μ)	0.4	0.4	0.4	0.6	0.6
Epoch Size (ε)	16	16	16	16	16
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	51	51	51	51	51
No. of Nodes in Hidden Layer 1 (N^{H1})	45	45	45	45	45
No. of Nodes in Hidden Layer 2 (N^{H2})	15	15	15	15	15
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	46	47	48	49	50
Learning Rate (η)	0.02	0.1	0.2	0.02	0.1
Momentum (μ)	0.8	0.8	0.8	0.9	0.9
Epoch Size (ε)	16	16	16	16	16
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	51	51	51	51	51
No. of Nodes in Hidden Layer 1 (N^{H1})	45	45	45	45	45
No. of Nodes in Hidden Layer 2 (N^{H2})	15	15	15	15	15
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	51	52	53	54	55
Learning Rate (η)	dbd: *)	dbd: *)	0.01	0.005	0.005
Momentum (μ)	0.005, 0.6	0.07, 0.4	0.6	0.6	0.6
Epoch Size (ε)	5	5	5	5	5
Transfer Function	Tanh	Tanh	Sigmoidal	Linear	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Cubic
No. of Inputs (N^I)	51	51	51	51	51
No. of Nodes in Hidden Layer 1 (N^{H1})	45	45	45	45	45
No. of Nodes in Hidden Layer 2 (N^{H2})	15	15	15	15	15
No. of Outputs (N^O)	1	1	1	1	1

*) Note: These parameters are different from the conventional learning rate and momentum values.

Parameter / Geometry	Model Number				
	56	57	58	59	60
Learning Rate (η)	0.005	0.005	0.005	0.005	0.005
Momentum (μ)	0.6	0.6	0.6	0.6	0.6
Epoch Size (ε)	5	5	5	5	5
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quartic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	51	51	51	51	51
No. of Nodes in Hidden Layer 1 (N^{H1})	45	45	45	45	45
No. of Nodes in Hidden Layer 2 (N^{H2})	15	15	10	20	25
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	61	62	63	64	65
Learning Rate (η)	0.005	0.005	0.005	0.005	0.005
Momentum (μ)	0.6	0.6	0.6	0.6	0.6
Epoch Size (ε)	5	5	5	5	5
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	51	51	51	51	51
No. of Nodes in Hidden Layer 1 (N^{H1})	15	30	60	90	15
No. of Nodes in Hidden Layer 2 (N^{H2})	5	10	20	30	0
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	66	67	68	69	70
Learning Rate (η)	0.005	0.005	0.005	0.005	0.1
Momentum (μ)	0.6	0.6	0.6	0.6	0.6
Epoch Size (ε)	5	5	5	5	16
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	51	51	51	51	51
No. of Nodes in Hidden Layer 1 (N^{H1})	30	45	60	90	30
No. of Nodes in Hidden Layer 2 (N^{H2})	0	0	0	0	0
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	71	72	73	74	75
Learning Rate (η)	0.1	0.001	0.005	0.005	0.1
Momentum (μ)	0.6	0	0.6	0.6	0.6
Epoch Size (ε)	16	16	5	5	16
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	51	51	51	51	51
No. of Nodes in Hidden Layer 1 (N^{H1})	60	45	60	90	30
No. of Nodes in Hidden Layer 2 (N^{H2})	20	15	0	0	0
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	76	77	78	79	80
Learning Rate (η)	Default	0.02	0.02	0.02	0.02
Momentum (μ)	Default	0.6	0.6	0.6	0.6
Epoch Size (ε)	16	16	16	16	5
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	51	51	51	39	56
No. of Nodes in Hidden Layer 1 (N^{H1})	45	45	45	30	60
No. of Nodes in Hidden Layer 2 (N^{H2})	15	15	0	0	20
No. of Outputs (N^O)	3	3	3	3	1

Parameter / Geometry	Model Number				
	81	82	83	84	85
Learning Rate (η)	0.002	0.1	0.1	0.01	0.01
Momentum (μ)	0.6	0.6	0.6	0.6	0.6
Epoch Size (ε)	16	16	16	16	16
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	56	41	30	30	50
No. of Nodes in Hidden Layer 1 (N^{H1})	50	45	45	45	60
No. of Nodes in Hidden Layer 2 (N^{H2})	0	15	15	15	20
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	86	87	88	89	90
Learning Rate (η)	0.02	0.02	0.02	0.02	0.02
Momentum (μ)	0.6	0.6	0.6	0.6	0.6
Epoch Size (ε)	16	16	16	16	16
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	47	47	47	25	25
No. of Nodes in Hidden Layer 1 (N^{H1})	15	20	35	5	15
No. of Nodes in Hidden Layer 2 (N^{H2})	0	0	0	0	0
No. of Outputs (N^O)	1	1	1	1	1

Parameter / Geometry	Model Number				
	91	92	93	94	95
Learning Rate (η)	0.02	0.02	0.02	0.02	
Momentum (μ)	0.6	0.6	0.6	0.6	
Epoch Size (ε)	16	16	16	16	
Transfer Function	Tanh	Tanh	Tanh	Tanh	
Learning Rule	NCDE	NCDE	NCDE	NCDE	
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	
No. of Inputs (N^I)	25	36	36	36	
No. of Nodes in Hidden Layer 1 (N^{H1})	30	15	20	35	
No. of Nodes in Hidden Layer 2 (N^{H2})	0	0	0	0	
No. of Outputs (N^O)	1	1	1	1	

D.2 Modelling Blue-Green Algae in the River Murray

Parameter / Geometry	Model Number				
	01	02	03	04	05
Learning Rate (η)	0.1	0.02	0.03	0.03	0.03
Momentum (μ)	0.6	0.6	0.6	0.6	0.6
Epoch Size (ε)	16	16	16	16	16
Transfer Function	Tanh	Tanh	Tanh	Tanh	Tanh
Learning Rule	NCDE	NCDE	NCDE	NCDE	NCDE
Error Function	Quadratic	Quadratic	Quadratic	Quadratic	Quadratic
No. of Inputs (N^I)	26	26	7	43	102
No. of Nodes in Hidden Layer 1 (N^{H1})	45	45	15	45	120
No. of Nodes in Hidden Layer 2 (N^{H2})	15	15	5	15	40
No. of Outputs (N^O)	1	1	2	2	2

Parameter / Geometry	Model Number				
	06	07			
Learning Rate (η)	0.03	0.03			
Momentum (μ)	0.6	0.6			
Epoch Size (ε)	16	16			
Transfer Function	Tanh	Tanh			
Learning Rule	NCDE	NCDE			
Error Function	Quadratic	Quadratic			
No. of Inputs (N^I)	7	43			
No. of Nodes in Hidden Layer 1 (N^{H1})	15	45			
No. of Nodes in Hidden Layer 2 (N^{H2})	5	15			
No. of Outputs (N^O)	2	2			

Appendix E

Neural Network Performance at Various Stages of Learning (Test Set)

E.1 One Day Forecasts

Learn Count	RMSE (EC Units)							
	Model SMB_U_A_XX_XX_91 - Differenced Data							
	04_02*)	05_02*)	06_02*)	07_02*)	08_02*)	09_01*)		
10,000	10.4	10.5	11.5	11.5	18.0	16.8		
20,000	10.3	10.4	10.8	10.8	17.8	12.1		
30,000	10.7	10.7	10.6	10.8	17.7	11.2		
40,000	10.3	10.4	10.7	10.6	17.3	11.1		
50,000	10.3	10.4	10.5	10.6	12.6	11.6		
60,000	10.3	10.4	10.5	11.0	10.9	10.7		
70,000	10.3	10.4	10.5	10.6	11.0	10.7		
80,000	10.4	10.5	10.5	10.6	10.7	10.8		
90,000	10.6	10.6	10.7	10.7	11.4	11.0		
100,000	10.3	10.4	10.6	11.0	10.7	10.8		

*) Note: These results are for the differenced data

Learn Count	RMSE (EC Units)							
	Model SMB_U_A_XX_XX_91							
	10_03	11_03	12_03	13_04				
50,00		37.7	42.5	28.3				
10,000	24.8	48.4	37.2	36.2				
15,000		34.7	31.3	39.2				
20,000	19.6	25.8	32.8	28.1				
25,000		32.2	30.6	28.2				
30,000	18.9	25.0	28.9	30.0				
35,000		26.7	24.4	27.1				
40,000	18.1	26.3	38.3	29.4				
45,000		31.5	28.3	26.2				
50,000	19.5	42.0	34.9	27.5				
55,000		32.6	33.9	26.0				
60,000	18.5	30.8	25.4	32.5				
65,000		43.9	26.8	25.1				
70,000	17.1	28.5	25.8	29.4				
75,000		31.5	31.1	24.9				
80,000	18.4	25.7	28.1	31.6				
85,000		25.0	29.8	25.5				
90,000	17.5	25.3	26.3	34.3				
95,000		31.9	25.3	27.5				
100,000	18.7	29.5	27.9	28.4				

E.2 14 Day Forecasts

Learn Count	RMSE (EC Units)							
	Model SMB_U_A_XX_XX_91							
	10_03	11_03	12_03	13_04	14_05			
5,000		112.3	110.6	104.2	119.2			
10,000	116.6	100.3	102.2	103.9	109.9			
11,500			97.9					
13,000			93.3					
14,500			87.9					
15,000		93.9	91.7	94.4	106.7			
16,000			99.8					
17,500			88.4					
19,000			134.7					
20,000	107.3	97.9	109.1	91.5	105.9			
22,000			98.6					
23,500			96.8					
25,000		127.2	98.8	101.5	107.4			
30,000	112.0	111.9	104.8	95.3	95.7			
35,000		104.8	113.7	92.4	106.7			
40,000	105.1	116.0	107.4	94.7	127.3			
45,000		112.9	105.8	95.2	120.3			
50,000	107.2	120.7	106.0	93.9	121.0			
55,000		104.0	105.3	93.1	127.0			
60,000	107.5	105.2	107.8	98.9	121.4			
65,000		109.5	115.1	96.2	125.6			
70,000	111.4	113.7	108.5	97.3	125.1			
75,000		126.6	119.8	92.6	122.3			
80,000	109.2	107.5	102.2	101.7	117.5			
85,000		113.4	108.5	92.3	116.9			
90,000	108.6	106.6	106.2	100.3	122.9			
95,000		117.1	120.3	97.6	119.1			
100,000	110.0	126.9	109.5	102.4	117.0			

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_XX_88				Model SMB_M_A_XX_XX_89			
	16_07	17_08	18_09	18_10	16_07	17_08	18_09	18_10
5,000		60.0	66.3	60.0		56.7	65.9	58.4
10,000	78.8	57.6	61.4	54.8	57.8	54.8	62.7	63.9
15,000		62.9	49.7	65.3		57.5	61.2	57.2
20,000	47.7	66.1	50.9	49.7	62.2	54.0	67.4	53.6
25,000		56.8	42.6	51.3		52.7	58.4	51.3
30,000	54.9	44.4	47.4	51.8	59.0	49.7	56.4	55.6
35,000		51.4	44.3	45.6		49.3	55.7	50.3
40,000	49.1	48.0	44.8	46.4	60.8	49.6	55.3	50.7
45,000		48.8	43.6	47.6		49.8	55.9	49.8
50,000	49.3	51.8	44.3	49.7	61.9	49.5	55.6	52.3
55,000		48.9		45.4		50.0	55.9	49.7
60,000		51.3		45.7		49.8	55.9	49.8
65,000		50.6		44.8		50.3	55.9	49.3
70,000		49.3				50.3		

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_XX_90				Model SMB_M_A_XX_XX_91			
	16_07	17_08	18_09	18_10	16_07	17_08	18_09	18_10
5,000		56.7	48.0	54.2		71.7	65.3	72.6
10,000	55.7	51.7	48.9	54.8	57.7	66.9	67.5	58.8
15,000		49.8	50.1	51.2		65.7	54.1	62.4
20,000	50.0	48.2	49.1	50.7	63.3	66.9	55.7	61.6
25,000		46.1	46.7	50.0		58.1	51.6	60.2
30,000	50.5	45.5	45.2	46.0	63.5	55.6	53.5	49.0
35,000		44.9	45.9	47.0		55.0	48.9	51.8
40,000	51.3	45.1	46.4	46.6	63.7	54.4	47.6	50.2
45,000		44.9	46.9	46.1		54.8	44.4	48.6
50,000	52.6	45.2		45.7	63.7	54.1	47.8	47.4
55,000				45.3		55.0	45.6	47.6
60,000		44.5		44.6		54.0	44.7	46.7
65,000				44.5		54.4	45.0	45.8
70,000		44.6				52.0		

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_19_11_XX							
	88	89	90	91				
5,000	53.9	61.1	48.8	56.8				
10,000	67.6	59.2	57.4	67.0				
15,000	44.5	54.3	56.2	62.7				
20,000	45.2	60.5	49.9	50.1				
25,000	42.9	52.8	46.4	55.4				
30,000	46.9	53.8	52.3	52.7				
35,000	41.0	50.1	51.4	51.8				
40,000	41.7	48.4	51.7	52.2				
45,000	41.3	49.4	49.3	48.8				
50,000	42.1	49.6	49.1	51.2				
55,000	41.5	49.0	49.3	52.2				
60,000		49.3	48.3	48.4				
65,000		53.0	49.7	49.0				

Model SMB_M_A_19_11_88					
Learn count	RMSE (EC units)	Learn count	RMSE (EC units)	Learn count	RMSE (EC units)
15,008	44.4	35,200	43.6	35,792	42.7
15,024	50.2	35,216	41.0	35,808	41.3
15,040	45.2	35,232	44.4	35,824	42.8
15,056	51.6	35,248	43.8	35,840	42.0
15,072	62.3	35,264	42.7	35,856	44.1
15,088	46.1	35,280	42.3	35,872	40.9
15,104	45.0	35,296	41.7	35,888	42.2
15,120	46.6	35,312	42.4		
15,136	47.2	35,328	42.5	55,008	41.5
15,152	48.8	35,344	41.7	55,024	42.4
15,168	45.6	35,360	41.7	55,040	42.0
15,184	57.4	35,376	42.1	55,056	43.3
15,200	48.7	35,392	41.3	55,072	48.0
15,216	47.7	35,408	40.8	55,088	44.6
15,232	63.4	35,424	42.0	55,104	41.7
15,248	53.1	35,440	42.9	55,120	42.1
15,280	46.9	35,456	40.7	55,136	41.9
15,296	45.0	35,472	42.9	55,152	42.8
15,312	51.3	35,488	42.1	55,168	42.4
15,328	46.1	35,504	41.8	55,184	44.6
15,344	45.2	35,520	41.6	55,200	44.1
15,360	44.0	35,536	42.8	55,216	41.4
15,376	47.2	35,552	42.2	55,232	45.8
15,392	44.1	35,568	42.3	55,248	44.6
		35,584	42.5	55,264	43.2
35,008	41.0	35,600	41.1	55,280	42.8
35,024	41.8	35,616	42.2	55,296	42.2
35,040	41.5	35,632	43.4	55,312	43.4
35,056	42.5	35,648	42.3	55,328	43.6
35,072	46.3	35,664	42.2	55,344	42.5
35,088	43.9	35,680	41.8	55,360	42.6
35,104	41.2	35,696	42.0	55,376	42.9
35,120	41.4	35,712	43.4	55,392	41.8
35,136	41.3	35,728	41.3		
35,152	42.0	35,744	42.4		
35,168	41.8	35,760	42.0		
35,184	43.6	35,776	41.7		

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88							
	20	20a	20b	21	22	23	24	25
5,000	59.2	61.9	61.8	61.6	62.7	63.5	65.0	65.3
10,000	57.3	62.3	56.4	59.9	61.6	63.1	64.1	64.9
15,000	58.2	55.1	57.8	59.4	59.9	61.0	62.6	63.7
20,000	55.2	55.4	55.0	56.1	56.2	60.1	60.5	63.1
25,000	53.0	53.9	55.3	54.8	55.1	56.8	61.4	63.6
30,000	53.3	52.6	58.8	54.3	54.5	57.1	59.1	61.7
35,000	53.4	51.4	53.0	53.5	54.6	55.9	58.8	61.8
40,000	54.2	53.5	50.3	53.6	53.9	56.0	57.4	59.6
45,000	48.7	52.6	47.9	51.7	52.8	54.5	58.3	59.6
50,000	51.1	46.6	47.8	54.0	53.8	54.4	56.1	59.1
55,000	47.2	47.6	48.0	51.2	53.2	56.7	57.6	58.6
60,000	47.0	46.6	46.5	49.6	51.2	54.2	58.1	60.4
65,000	45.9	43.8	48.0	49.3	52.1	54.4	58.4	58.1
70,000	45.7	43.8	43.3	48.7	51.9	53.5	55.2	57.4
75,000	43.5	43.5	46.0	46.7	49.4	53.3	55.2	56.3
80,000	46.1	45.0	44.5	47.2	50.1	51.9	54.3	56.1
85,000	43.0	44.2	42.9	45.3	48.1	53.0	54.7	56.8
90,000	45.0	42.3	42.7	45.2	47.2	52.2	55.7	56.0
95,000	44.4	46.0	43.5	48.9	50.3	53.6	53.7	55.9
100,000	42.2	43.8	43.0	43.8	45.5	50.2	54.7	55.2
105,000	42.0			44.0	46.7	50.7	53.6	56.7
110,000	43.8			44.1	45.4	49.9	53.1	56.0
115,000	43.8			44.3	46.4	48.4	53.7	55.6
120,000	45.0			44.7	46.4	50.2	53.4	55.6
125,000	42.0			42.5	44.3	48.6	53.6	55.2
130,000	46.3			43.9	45.3	48.2	53.3	53.8
135,000	43.1			42.6	43.7	48.1	52.9	54.8
140,000	45.6			42.9	43.3	47.3	51.8	54.9
145,000	44.3			45.0	46.4	46.9	53.5	54.4
150,000	42.6			42.2	43.0	45.8	54.1	53.6
155,000	42.2			44.7	42.9	47.4	54.5	55.2
160,000	44.3			42.6	43.2	45.6	53.0	55.0
165,000	44.56			42.9	43.9	46.9	51.2	54.8
170,000	46.2			43.3	44.6	45.6	49.3	52.8
175,000	43.2			43.9	42.2	46.0	49.1	53.1

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88 (cont.)							
	20	20a	20b	21	22	23	24	25
180,000	47.0			43.2	43.8	44.9	49.4	52.8
185,000	44.4			45.6	42.9	46.1	50.1	52.7
190.000	47.0			42.2	42.4	44.6	48.7	53.8
195,000	45.6			43.5	45.3	45.7	50.9	52.2
200,000	44.0			42.2	42.7	43.7	50.0	51.8
205,000	47.1					44.2	47.6	52.5
210,000	50.0					43.8	48.9	54.1
215,000	46.8					44.1	46.4	53.2
220,000	48.7					43.4	47.9	50.6
225,000	47.7					43.3	47.6	50.7
230,000	47.7					43.5	47.4	51.2
235,000	48.5					43.9	46.8	51.5
240,000	48.9					44.0	47.0	51.6
245,000	50.4					44.1	45.7	50.6
250,000	48.6					44.1	45.8	49.0
255,000						43.4	45.7	50.0
260,000						42.5	49.2	50.4
265,000						44.5	45.8	48.5
270,000						44.4	45.4	48.9
275,000						42.7	46.1	51.5
280,000						44.8	44.8	50.1
285,000						42.5	44.7	48.4
290,000						44.7	46.6	49.6
295,000						42.6	45.0	49.6
300,000						44.3	46.4	49.2
305,000						43.8	44.6	48.6
310,000						45.0	44.3	47.6
315,000						45.5	43.8	47.3
320,000						42.3	43.9	47.8
325,000						45.0	43.3	47.3
330,000						43.1	43.4	48.4
335,000						43.6	47.0	46.8
340,000						43.9	44.9	46.5
345,000						42.9	43.0	48.8
350,000						44.2	43.1	47.3

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88 (cont.)							
	20	20a	20b	21	22	23	24	25
355,000							43.7	46.3
360,000							43.0	46.6
365.000							42.8	49.5
370,000							43.4	45.4
375,000							44.2	45.1
380,000							43.5	45.6
385,000							45.4	45.4
390,000							43.6	47.6
395,000							42.5	45.2
400,000							42.4	45.1
405,000							44.6	46.4
410,000							44.6	44.5
415,000							44.3	44.4
420,000							43.9	44.1
425,000							42.5	44.4
430,000							42.4	46.0
435,000							43.5	43.8
440,000							43.3	45.7
445,000							43.8	46.0
450,000							43.1	44.8
455,000							42.5	43.6
460,000							43.8	43.4
465,000							43.4	44.7
470,000							42.7	46.1
475,000							42.6	44.8
480,000							43.5	43.9
485,000							44.7	44.1
490,000							44.9	43.1
495,000							43.4	43.3
500,000							42.7	44.6
505,000								43.7
510,000								47.0
515,000								45.1
520,000								44.4
525,000								43.0

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88 (cont.)							
	20	20a	20b	21	22	23	24	25
530,000								43.9
535,000								44.3
540.000								43.9
545,000								43.2
550,000								46.0
555,000								42.6
560,000								43.9
565,000								43.8
570,000								44.7
575,000								44.0
580,000								44.0
585,000								44.1
590,000								44.9
595,000								43.3
600,000								43.9

Model SMB_M_A_26_10_88					
Learn count	RMSE (EC units)	Learn count	RMSE (EC units)	Learn count	RMSE (EC units)
5,000	69.4	80,000	58.9	155,000	57.1
10,000	65.9	85,000	58.6	160,000	54.6
15,000	64.8	90,000	56.7	165,000	55.1
20,000	64.4	95,000	57.3	170,000	56.3
25,000	63.3	100,000	56.8	175,000	54.8
30,000	62.9	105,000	56.0	180,000	55.0
35,000	63.4	110,000	58.6	185,000	54.9
40,000	61.6	115,000	56.5	190,000	55.6
45,000	61.0	120,000	56.2	195,000	54.9
50,000	60.5	125,000	55.1	200,000	55.8
55,000	59.9	130,000	57.1	205,000	53.9
60,000	61.2	135,000	56.9	210,000	53.3
65,000	58.7	140,000	55.3	215,000	53.3
70,000	58.6	145,000	56.0	220,000	53.7
75,000	57.6	150,000	55.6	225,000	53.6

Model SMB_M_A_26_10_88 (cont.)					
Learn count	RMSE (EC units)	Learn count	RMSE (EC units)	Learn count	RMSE (EC units)
230,000	52.6	400,000	53.6	570,000	46.4
235,000	53.2	405,000	51.8	575,000	45.0
240,000	53.6	410,000	47.8	580,000	44.8
245,000	53.8	415,000	49.4	585,000	44.5
250,000	52.5	420,000	49.8	590,000	45.2
255,000	52.3	425,000	49.4	595,000	45.4
260,000	52.6	430,000	47.5	600,000	46.1
265,000	53.5	435,000	50.1	605,000	44.1
270,000	53.1	440,000	49.2	610,000	44.5
275,000	54.2	445,000	49.1	615,000	44.7
280,000	52.3	450,000	48.2	620,000	44.8
285,000	51.9	455,000	48.5	625,000	44.6
290,000	53.1	460,000	46.9	630,000	44.6
295,000	52.3	465,000	47.2	635,000	43.6
300,000	53.0	470,000	50.1	640,000	44.5
305,000	51.5	475,000	46.1	645,000	44.1
310,000	51.7	480,000	46.2	650,000	44.0
315,000	52.1	485,000	46.1	655,000	44.6
320,000	51.1	490,000	46.4	660,000	44.0
325,000	50.7	495,000	46.4	665,000	43.3
330,000	51.5	500,000	49.8	670,000	43.8
335,000	50.9	505,000	47.9	675,000	43.2
340,000	51.2	510,000	48.4	680,000	44.6
345,000	52.4	515,000	46.8	685,000	43.1
350,000	50.6	520,000	48.5	690,000	44.4
355,000	50.7	525,000	45.5	695,000	43.3
360,000	50.6	530,000	45.1	700,000	43.6
365,000	49.1	535,000	46.3	705,000	42.9
370,000	51.2	540,000	46.3	710,000	43.8
375,000	51.0	545,000	45.3	715,000	43.0
380,000	48.4	550,000	45.1	720,000	43.5
385,000	49.8	555,000	46.0	725,000	43.5
390,000	48.4	560,000	44.5	730,000	43.2
395,000	50.4	565,000	45.3	735,000	44.6

Model SMB_M_A_26_10_88 (cont.)					
Learn count	RMSE (EC units)	Learn count	RMSE (EC units)	Learn count	RMSE (EC units)
740,000	43.1	795,000	43.4	850,000	42.5
745,000	44.9	800,000	42.5	855,000	42.8
750,000	42.9	805,000	44.2	860,000	42.7
755,000	43.5	810,000	43.5	865,000	43.2
760,000	42.7	815,000	43.9	870,000	46.3
765,000	43.5	820,000	43.5	875,000	43.1
770,000	43.9	825,000	43.0	880,000	42.8
775,000	42.6	830,000	43.2	885,000	42.9
780,000	44.9	835,000	44.0	890,000	43.7
785,000	44.2	840,000	46.0	895,000	42.8
790,000	44.1	845,000	42.6	900,000	44.3

Model SMB_M_A_27_10_88					
Learn count	RMSE (EC units)	Learn count	RMSE (EC units)	Learn count	RMSE (EC units)
16,710	80.0	601,560	51.9	1,269,960	47.3
33,420	65.6	634,980	52.4	1,303,380	43.9
50,130	64.6	668,400	64.3	1,336,800	43.6
66,840	63.3	701,820	51.9	1,370,220	44.0
83,550	62.2	735,240	49.8	1,403,640	47.1
100,260	62.1	768,660	49.6	1,437,060	44.3
116,970	61.1	802,080	48.7	1,470,480	50.9
167,100	59.1	835,500	53.9	1,503,900	43.3
200,520	57.6	868,920	49.9	1,537,320	43.6
233,940	56.7	902,340	49.3	1,570,740	42.7
267,360	56.7	935,760	49.6	1,604,160	44.2
300,780	55.6	969,180	47.2	1,637,580	43.2
334,200	56.2	1,002,600	47.5	1,671,000	51.2
367,620	56.0	1,036,020	48.1	1,704,420	43.1
401,040	53.5	1,069,440	46.9	1,737,840	43.4
434,460	52.9	1,102,860	48.5	1,771,260	43.0
467,880	55.2	1,136,280	48.1	1,804,680	50.4
501,300	55.4	1,169,700	47.2	1,838,100	44.1
534,720	56.8	1,203,120	47.3	1,871,520	42.3
568,140	55.6	1,236,540	44.1	1,904,940	45.6

Model SMB_M_A_27_10_88 (cont.)					
Learn count	RMSE (EC units)	Learn count	RMSE (EC units)	Learn count	RMSE (EC units)
1,938,360	45.8	1,971,780	44.4	2,005,200	48.2

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88							
	28	29	30	31	32			
5,000	58.7	61.7	59.4	53.2	62.3			
10,000	50.5	50.1	56.4	55.0	61.7			
15,000	42.9	51.4	50.6	47.3	62.0			
20,000	41.0	45.7	47.2	56.6	58.6			
25,000	42.1	44.8	51.7	47.9	56.0			
30,000	41.1	48.8	46.5	42.6	56.4			
35,000	44.8	43.6	45.4	44.7	56.6			
40,000	44.4	46.2	60.3	43.5	56.1			
45,000	43.1	45.5	39.8	44.2	54.3			
50,000	58.2	43.7	45.4	48.1	56.3			
55,000	57.3	39.9	60.9	44.6	53.5			
60,000	45.3	49.3	48.5	45.5	53.5			
65,000	44.8	49.2	46.1	46.3	53.0			
70,000	64.8	47.9	49.8	54.5	52.4			
75,000	53.3	50.7	50.5	45.7	52.3			
80,000	47.7	60.9	55.7	50.6	52.1			
85,000	62.6	61.2	48.8	52.3	52.3			
90,000	49.7	55.5	47.7	47.9	53.5			
95,000	46.6	49.4	53.0	51.6	53.0			
100,000	67.6	44.3	55.8	48.8	50.3			
105,000	50.2	43.7	54.1	52.6	48.3			
110,000	58.0	45.6	46.7	49.9	47.8			
115,000	58.6	54.7	51.9	64.2	47.4			
120,000	50.5	53.6	51.7	59.5	47.2			
125,000	56.3	51.5	49.9	56.9	46.4			
130,000	49.0	55.9	52.7	55.6	50.6			
135,000	54.2	55.4	47.8	58.6	47.0			
140,000	63.4	47.3	49.4	73.2	46.6			
145,000	65.0	60.1	52.0	55.7	46.4			
150,000	57.3	57.1	50.1	58.0	48.4			

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88 (cont.)							
	28	29	30	31	32			
155,000					45.1			
160,000					46.8			
165,000					44.3			
170,000					48.0			
175,000					44.4			
180,000					44.9			
185,000					43.5			
190.000					43.5			
195,000					44.7			
200,000					43.7			
205,000					42.9			
210,000					43.7			
215,000					43.7			
220,000					43.7			
225,000					42.3			
230,000					43.3			
235,000					43.0			
240,000					43.8			
245,000					44.3			
250,000					42.8			
255,000					43.3			
260,000					44.1			
265,000					44.9			
270,000					42.7			
275,000					42.9			
280,000					44.7			
285,000					43.1			
290,000					43.3			
295,000					46.2			
300,000					43.3			

Model SMB_M_A_33_10_88					
Learn count	RMSE (EC units)	Learn count	RMSE (EC units)	Learn count	RMSE (EC units)
10,000	63.2	360,000	55.5	710,000	51.6
20,000	64.3	370,000	54.9	720,000	50.6
30,000	64.6	380,000	54.8	730,000	50.4
40,000	63.7	390,000	54.8	740,000	50.6
50,000	63.6	400,000	55.1	750,000	51.5
60,000	63.2	410,000	55.3	760,000	53.8
70,000	63.0	420,000	54.7	770,000	56.3
80,000	62.5	430,000	55.0	780,000	50.1
90,000	62.0	440,000	54.4	790,000	50.5
100,000	61.4	450,000	55.4	800,000	50.7
110,000	60.9	460,000	54.0	810,000	49.3
120,000	60.7	470,000	54.5	820,000	53.4
130,000	60.0	480,000	54.7	830,000	49.6
140,000	59.4	490,000	55.2	840,000	53.2
150,000	59.8	500,000	55.6	850,000	53.9
160,000	59.4	510,000	54.6	860,000	52.5
170,000	58.1	520,000	54.7	870,000	50.7
180,000	58.6	530,000	52.7	880,000	49.3
190,000	58.1	540,000	54.5	890,000	50.5
200,000	57.7	550,000	53.1	900,000	48.3
210,000	58.0	560,000	52.2	910,000	47.9
220,000	57.4	570,000	53.3	920,000	50.8
230,000	56.2	580,000	52.3	930,000	49.3
240,000	56.3	590,000	53.2	940,000	48.8
250,000	56.7	600,000	53.6	950,000	48.5
260,000	56.9	610,000	54.0	960,000	49.0
270,000	56.8	620,000	51.9	970,000	54.6
280,000	56.1	630,000	51.7	980,000	47.8
290,000	55.4	640,000	52.8	990,000	48.4
300,000	56.0	650,000	53.4	1,000,000	51.5
310,000	55.8	660,000	52.5	1,010,000	46.6
320,000	56.5	670,000	52.7	1,020,000	47.1
330,000	55.9	680,000	53.5	1,030,000	49.0
340,000	55.9	690,000	52.7	1,040,000	61.7
350,000	55.5	700,000	53.7	1,050,000	46.4

Model SMB_M_A_33_10_88 (cont.)					
Learn count	RMSE (EC units)	Learn count	RMSE (EC units)	Learn count	RMSE (EC units)
1,060,000	46.5	1,410,000	43.8	1,760,000	42.9
1,070,000	53.0	1,420,000	46.7	1,770,000	43.6
1,080,000	46.8	1,430,000	44.2	1,780,000	42.7
1,090,000	49.6	1,440,000	46.5	1,790,000	43.0
1,100,000	49.7	1,450,000	45.3	1,800,000	43.4
1,110,000	48.2	1,460,000	43.7	1,810,000	43.5
1,120,000	45.9	1,470,000	43.7	1,820,000	44.0
1,130,000	51.8	1,480,000	47.7	1,830,000	42.8
1,140,000	45.8	1,490,000	45.1	1,840,000	42.4
1,150,000	45.7	1,500,000	43.4	1,850,000	43.7
1,160,000	50.0	1,510,000	43.9	1,860,000	43.2
1,170,000	45.8	1,520,000	46.1	1,870,000	44.1
1,180,000	46.8	1,530,000	44.9	1,880,000	45.7
1,190,000	47.1	1,540,000	44.8	1,890,000	46.4
1,200,000	45.2	1,550,000	43.3	1,900,000	42.5
1,210,000	45.1	1,560,000	46.3		
1,220,000	45.0	1,570,000	43.7		
1,230,000	46.0	1,580,000	43.7		
1,240,000	45.6	1,590,000	43.9		
1,250,000	45.8	1,600,000	43.1		
1,260,000	46.7	1,610,000	46.9		
1,270,000	49.1	1,620,000	43.5		
1,280,000	44.7	1,630,000	43.0		
1,290,000	45.1	1,640,000	44.0		
1,300,000	44.5	1,650,000	47.4		
1,310,000	44.5	1,660,000	42.8		
1,320,000	52.3	1,670,000	43.1		
1,330,000	44.3	1,680,000	42.8		
1,340,000	44.6	1,690,000	43.9		
1,350,000	44.4	1,700,000	43.1		
1,360,000	44.2	1,710,000	44.1		
1,370,000	44.1	1,720,000	43.1		
1,380,000	45.7	1,730,000	43.8		
1,390,000	45.2	1,740,000	42.6		
1,400,000	49.8	1,750,000	44.5		

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88							
	34	35	36	37	38	39	40	41
5,000	63.5	63.1	61.6	58.9	59.4	53.2	55.7	56.7
10,000	63.8	62.9	57.0	53.6	58.7	57.1	58.5	56.3
15,000	63.3	61.9	58.2	56.8	57.4	56.1	59.8	57.2
20,000	61.8	59.4	53.1	49.1	56.4	54.7	60.9	56.7
25,000	59.4	56.7	53.4	46.2	53.1	57.5	99.6	50.9
30,000	59.6	56.7	51.1	45.0	54.1	48.9	65.0	50.6
35,000	58.9	56.2	50.4	44.6	54.7	54.7	58.9	50.7
40,000	57.6	55.7	48.8	45.6	59.1	67.8	62.6	55.2
45,000	56.3	54.4	46.9	42.2	50.3	42.7	44.1	44.6
50,000	56.5	55.2	47.5	43.4	51.9	41.5	47.0	47.4
55,000	57.2	54.9	47.6	44.7	47.1	55.1	48.1	43.6
60,000	55.7	53.5	44.8	44.3	46.3	42.8	62.4	43.6
65,000	56.0	54.2	45.1	42.9	48.5	46.7	57.4	45.2
70,000	55.7	54.3	45.0	44.2	47.0	54.6	57.5	44.0
75,000	54.9	52.8	42.8	43.0	44.8	42.2	53.3	42.8
80,000	54.8	52.9	44.6	44.2	43.5	58.6	65.6	41.9
85,000	54.5	52.0	44.3	47.1	46.4	46.3	62.0	45.9
90,000	54.5	51.8	42.4	44.3	43.0	42.4	54.5	42.6
95,000	55.3	53.6	45.3	46.9	52.9	43.8	53.1	53.5
100,000	53.5	50.2	42.6	45.9	42.9	43.9	59.3	42.6
105,000	54.2	51.9	43.8	48.1	43.2	43.9	53.3	45.8
110,000	53.5	49.7	43.8	50.4	42.1	44.5	61.0	43.0
115,000	53.6	49.3	46.6	50.5	42.0	44.1	53.7	42.5
120,000	53.5	48.9	42.5	46.4	47.4	51.6	48.3	48.0
125,000	52.5	49.0	43.6	48.3	42.3	44.1	54.6	44.7
130,000	52.9	47.9	44.8	52.2	42.9	53.4	57.7	46.6
135,000	52.0	47.7	45.4	48.3	46.3	47.6	47.2	48.3
140,000	52.3	50.0	45.0	51.2	43.5	51.8	58.7	45.8
145,000	53.6	50.5	44.3	52.4	43.5	47.8	46.2	45.6
150,000	50.9	47.2	44.6	48.7	42.7	64.9	49.1	45.8
155,000	51.7	49.2			46.0			48.1
160,000	51.2	46.3			44.7			45.2
165,000	51.0	47.3			42.1			45.3
170,000	50.8	46.1			46.3			50.4
175,000	49.4	46.3			44.7			49.7

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88 (cont.)							
	34	35	36	37	38	39	40	41
180,000	50.5	46.9			44.4			48.2
185,000	48.8	44.7			42.9			45.9
190.000	49.3	44.5			44.2			47.5
195,000	50.9	46.3			46.3			52.8
200,000	47.7	43.8			50.1			54.6
205,000	47.6	43.7			46.8			
210,000	48.6	43.8			47.2			
215,000	49.2	44.2			45.7			
220,000	49.2	44.5			45.8			
225,000	47.9	42.9			45.9			
230,000	47.9	43.9			46.0			
235,000	49.0	42.9			48.9			
240,000	46.9	42.8			45.6			
245,000	46.5	45.2			48.0			
250,000	45.7	42.8			47.0			
255,000	46.3	42.7			54.4			
260,000	45.1	42.8			51.1			
265,000	46.6	43.6			50.9			
270,000	46.3	42.6			49.2			
275,000	44.8	43.2			53.8			
280,000	44.8	43.6			47.0			
285,000	44.7	43.2			49.7			
290,000	44.7	44.7			47.3			
295,000	45.3	45.0			51.6			
300,000	44.2	43.0			48.7			
305,000	46.7							
310,000	45.1							
315,000	43.6							
320,000	44.3							
325,000	44.8							
330,000	44.3							
335,000	43.7							
340,000	43.5							
345,000	44.8							
350,000	44.6							

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88 (cont.)							
	34	35	36	37	38	39	40	41
355,000	43.2							
360,000	44.3							
365.000	43.9							
370,000	43.1							
375,000	44.3							
380,000	45.1							
385,000	42.9							
390,000	44.4							
395,000	42.5							
400,000	44.4							
405,000	42.6							
410,000	43.4							
415,000	43.3							
420,000	44.0							
425,000	42.4							
430,000	43.5							
435,000	42.8							
440,000	43.2							
445,000	44.5							
450,000	42.9							
455,000	42.3							
460,000	42.8							
465,000	43.6							
470,000	42.6							
475,000	43.4							
480,000	44.2							
485,000	43.2							
490,000	44.9							
495,000	45.7							
500,000	43.0							

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88							
	42	43	44	45	46	47	48	49
5,000	56.0	53.1	54.5	69.5	54.6	56.3	386.7	51.1
10,000	55.5	56.4	54.3	77.8	51.8	56.5	522.0	43.8
15,000	60.0	56.8	57.1	54.3	50.6	63.2	387.1	42.4
20,000	48.1	53.1	55.0	47.8	47.4	57.3	559.3	48.0
25,000	52.6	80.6	46.3	69.4	45.0	53.2	389.3	46.2
30,000	42.3	43.8	46.1	39.5	46.3	56.9	269.2	48.6
35,000	45.8	50.1	49.9	60.1	47.5	56.1	385.4	48.9
40,000	54.7	46.5	46.2	52.5	43.1	69.8		46.2
45,000	40.1	44.8	42.6	45.8	41.6	40.6		43.4
50,000	46.5	44.5	45.5	49.0	48.3	52.9		53.8
55,000	41.7	42.9	42.9	57.1	44.2	48.8		47.4
60,000	46.3	53.4	44.1	66.5	45.6	62.5		45.2
65,000	43.5	51.9	45.8	62.5	47.8	63.9		52.7
70,000	45.4	51.2	46.0	55.2	48.1	54.9		49.3
75,000	46.4	52.4	43.5	51.2	45.6	53.5		52.0
80,000	56.9	59.3	42.7	55.4	44.7	48.3		48.4
85,000	56.4	56.7	47.6	56.9	44.8	56.2		49.3
90,000	51.8	52.4	45.2	55.9	47.1	62.9		50.6
95,000	49.8	51.5	54.8	56.6	51.6	58.1		50.1
100,000	45.5	39.3	44.6	53.5	46.7	52.9		48.3
105,000	44.2	43.5	50.9	68.8	51.4	48.3		54.1
110,000	55.3	43.6	46.2	69.5	45.7	53.5		49.6
115,000	46.5	68.8	45.3	55.0	45.1	50.9		44.2
120,000	51.1	52.3	51.2	53.3	56.1	64.8		55.8
125,000	59.7	44.9	51.0	55.5	51.6	54.8		51.0
130,000	51.8	56.1	52.0	54.6	54.9	53.5		56.7
135,000	54.3	52.7	53.7	62.7	55.1	55.6		56.0
140,000	45.3	44.7	49.5	61.6	52.5	48.0		51.8
145,000	53.7	53.3	48.9	68.8	49.3	59.7		49.8
150,000	52.5	44.1	49.7	67.1	53.9	50.9		56.2

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88							
	50	51	52	53	54	55	56	
5,000	385.4	58.3	60.8	88.6	62.8	67.9	68.8	
10,000	400.4	57.0	55.1	60.2	63.8	71.7	65.1	
15,000		54.5	43.1	70.2	64.3	71.9	65.4	
20,000		51.6	43.9	61.2	59.8	72.6	62.4	
25,000		51.3	56.3	61.7	58.1	71.6	63.5	
30,000		49.7	48.7	64.2	60.2	69.5	62.6	
35,000		49.7	53.0	62.6	58.1	71.8	60.3	
40,000		48.8	49.1	64.7	60.0	70.1	60.5	
45,000		47.1	57.3	61.8	56.4	69.4	59.8	
50,000		47.5	54.9	69.1	61.7	69.2	58.7	
55,000		45.2	56.9	63.7	55.6	69.3	61.7	
60,000		45.1	65.8	63.3	55.1	69.4	59.8	
65,000		44.7	43.3	62.5	54.7	69.0	58.5	
70,000		46.1	62.0	63.6	54.5	69.1	55.8	
75,000		44.3	51.0	65.0	53.5	68.8	54.8	
80,000		44.6	48.0	64.1	56.3	68.6	55.1	
85,000		44.7	52.8	62.7	56.5	67.9	55.8	
90,000		45.3	58.7	62.8	56.8	67.2	55.4	
95,000		48.5	62.3	64.1	56.4	66.6	55.8	
100,000		46.5	47.5	62.7	54.8	66.4	53.3	
105,000		44.4	54.9	63.4	52.6	65.0	54.9	
110,000		44.2	54.5	63.6	52.4	66.4	53.8	
115,000		42.4	55.3	63.1	52.5	65.8	54.6	
120,000		47.0	63.6	62.1	53.7	65.5	54.2	
125,000		47.9	54.3	62.0	52.1	65.1	55.3	
130,000		46.1	52.9	61.1	53.2	64.9	53.0	
135,000		47.6	55.9	60.9	53.5	65.1	53.9	
140,000		44.6	62.9	62.2	51.8	64.2	56.2	
145,000		46.4	48.6	59.7	53.2	63.6	55.7	
150,000		50.4	54.3	59.3	55.4	63.4	55.0	
155,000		46.0	51.1	58.6	50.4	62.7	54.7	
160,000		44.3	64.4	59.4	53.1	63.0	54.8	
165,000		43.0	56.5	57.5	51.5	63.9	53.0	
170,000		51.2	63.4	56.7	53.4	62.0	51.9	
175,000		47.4	51.2	56.1	51.5	63.2	50.7	

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88 (cont.)							
	50	51	52	53	54	55	56	
180,000		49.4	57.7	56.6	50.8	63.0	51.3	
185,000		44.5	56.8	55.4	51.2	61.9	51.3	
190.000		45.1	49.4	54.6	51.6	63.2	51.4	
195,000		49.8	62.2	56.0	55.1	61.7	52.7	
200,000		53.3	60.3	56.6	60.5	61.8	49.6	
205,000				56.1	52.1			
210,000				56.9	55.6		52.7	
215,000				53.3	53.9			
220,000				55.4	51.1	60.1	49.7	
225,000				58.7	51.5			
230,000				53.3	52.6		49.4	
235,000				53.0	51.5			
240,000				54.4	52.6	59.7	49.4	
245,000				52.7	54.5			
250,000				52.8	52.4		47.7	
260,000				53.2		58.3	47.8	
270,000							48.2	
280,000				52.8		59.2	47.9	
290,000							50.2	
300,000				51.3		60.4	49.1	
310,000							47.6	
320,000				51.4		59.1	49.4	
330,000							47.2	
340,000				52.0		58.0	47.3	
350,000							46.9	
360,000				50.1		58.2	48.8	
370,000							47.4	
380,000				49.1		57.3	48.7	
390,000							47.1	
400,000				49.7		57.4	45.4	
410,000							48.5	
420,000				48.0		55.7	47.4	
430,000							46.2	
440,000				47.0		56.0	47.1	
450,000							46.2	

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88 (cont.)							
	50	51	52	53	54	55	56	
460,000				46.5		56.0	49.0	
470,000							46.9	
480.000				46.6		55.5	47.3	
490,000							47.4	
500,000				48.0		55.1	46.6	
510,000							46.6	
520,000				45.1		55.4	47.0	
530,000							47.5	
540,000				45.2		56.0	49.0	
550,000							48.0	
560,000				44.9		56.3		
580,000				44.5		54.0		
600,000				43.9		54.6		
620,000				44.3		54.8		
640,000				48.2		54.5		
660,000				43.1		54.7		
680,000				45.1		53.6		
700,000				43.8		54.2		
720,000				43.2		53.7		
740,000				44.9		55.3		
760,000				44.9		53.1		
780,000				43.2		53.6		
800,000				43.0		54.0		
820,000				43.9		52.8		
840,000				43.3		53.1		
860,000				44.3		51.9		
880,000				45.6		53.1		
900,000				44.1		54.0		
920,000						52.1		
940,000						52.4		
960,000						51.2		
980,000						52.4		
1,000,000						51.7		

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88 (cont.)							
	50	51	52	53	54	55	56	
1,020,000						52.5		
1,040,000						51.7		
1,060,000						52.1		
1,080,000						51.5		
1,100,000						53.3		
1,120,000						50.5		
1,140,000						51.2		
1,160,000						51.0		
1,180,000						50.3		
1,200,000						50.2		
1,220,000						50.7		
1,240,000						49.5		
1,260,000						51.7		
1,280,000						49.6		
1,300,000						50.0		

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88							
	57	58	59	60	61	62	63	64
10,000	60.5	55.6	52.8	55.4	63.6	58.6	56.3	55.9
20,000	55.8	54.1	59.9	53.0	60.9	56.9	52.2	53.7
30,000	54.5	53.3	54.8	55.2	60.4	55.8	51.1	57.7
40,000	48.9	52.6	51.4	51.3	52.2	48.9	49.9	50.3
50,000	48.8	50.2	48.6	51.8	51.5	48.2	54.9	48.0
60,000	48.4	45.9	45.1	47.4	47.7	46.3	50.2	44.6
70,000	47.3	45.3	44.5	45.0	47.0	47.1	42.3	46.5
80,000	44.2	44.0	47.6	46.5	47.5	45.3	48.5	43.2
90,000	43.9	42.4	43.6	45.5	46.0	44.0	41.9	45.4
100,000	43.4	45.5	44.3	47.1	48.1	45.5	43.3	43.8
110,000	45.8	47.9	47.0	45.3	45.6	46.9	44.6	42.4
120,000	45.3	43.6	45.0	42.9	47.8	49.2	44.5	42.6
130,000	44.5	44.2	45.3	43.4	44.9	49.4	44.3	46.9
140,000	45.7	44.5	49.2	44.1	47.7	47.8	46.3	46.6
150,000	45.5	45.3	48.1	43.0	47.7	45.2	46.2	47.4
160,000	45.3	44.0	46.1	45.1	45.7	47.3	48.2	48.0
170,000	47.8	51.4	47.9	45.4	47.1	45.9	46.7	47.7
180,000	45.2	44.8	47.2	43.4	49.9	47.6	44.6	50.3
190,000	45.0	46.7	47.6	46.2	45.2	49.3	47.9	51.8
200,000	48.0	47.5	48.9	45.4	47.2	48.4	47.2	51.3

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88							
	65	66	67	68	69			
10,000	54.5	52.8	52.8	53.9	50.8			
20,000	50.6	51.6	51.5	52.6	50.7			
30,000	51.8	54.5	52.2	53.6	50.8			
40,000	51.3	54.5	55.9	56.3	48.2			
50,000	47.2	48.5	56.0	54.9	47.3			
60,000	43.0	45.4	55.2	50.1	44.8			
70,000	43.5	43.9	44.8	49.9	41.7			
80,000	42.6	43.4	47.9	43.4	43.1			
90,000	45.6	40.9	45.0	42.8	41.8			
100,000	48.4	41.7	47.2	43.4	42.9			
110,000	46.5	42.6	45.4	44.1	43.1			
120,000	45.6	43.5	44.9	43.3	42.4			
130,000	44.3	43.3	57.0	44.0	42.4			
140,000	46.9	44.9	48.5	47.6	43.5			
150,000	45.4	42.3	49.1	48.9	42.0			
160,000	49.2	43.8	51.8	44.8	42.9			
170,000	49.1	43.1	46.2	47.9	45.0			
180,000	49.7	46.7	46.3	44.7	44.1			
190,000	46.1	48.4	48.2	47.4	47.4			
200,000	47.9	47.0	53.5	47.6	46.5			

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_10_88							
	70	71						
5,000	53.3	54.5						
10,000	48.9	48.7						
15,000	54.7	58.1						
20,000	40.9	43.7						
25,000	45.6	42.6						
30,000	51.8	42.1						
35,000	40.9	39.0						
40,000	51.8	49.6						
45,000	50.0	55.0						
50,000	40.4	46.6						
55,000	51.2	56.0						
60,000	44.1	60.5						
65,000	46.9	55.0						
70,000	50.7	44.8						
75,000	77.9	53.3						
80,000	43.5	66.0						
85,000	43.4	52.2						
90,000	43.5	49.5						
95,000	47.0	48.3						
100,000	45.3	51.4						

Model SMB_M_A_71_10_88					
Learn count	RMSE (EC units)	Learn count	RMSE (EC units)	Learn count	RMSE (EC units)
1,671	56.9	18,381	48.5	35,091	42.7
3,342	65.8	20,052	43.7	36,762	44.5
5,013	64.6	21,723	41.3	38,433	53.1
6,684	53.6	23,394	45.6	40,104	43.3
8,355	55.2	25,065	44.9	41,775	43.3
10,026	48.0	26,736	42.6	43,446	46.5
11,697	44.8	28,407	45.5	45,117	49.9
13,368	42.9	30,078	40.6	46,778	59.7
15,039	59.1	31,749	42.6	48,449	47.8
16,710	42.9	33,420	46.4	50,120	48.7

Model SMB_M_A_72_10_88					
Learn count	RMSE (EC units)	Learn count	RMSE (EC units)	Learn count	RMSE (EC units)
1,000	157.2	500,000	54.4	1,800,000	44.1
3,000	134.5	600,000	54.0	2,000,000	43.3
6,000	69.8	700,000	53.3	2,200,000	43.6
7,000	61.6	800,000	52.1	2,400,000	43.0
9,000	61.4	900,000	50.4	2,600,000	44.1
11,000	62.9	1,000,000	49.4	2,800,000	43.8
12,000	63.3	1,100,000	48.4	3,000,000	43.2
50,000	63.1	1,200,000	48.4	3,200,000	43.8
100,000	61.5	1,300,000	46.9	3,400,000	43.7
200,000	58.7	1,400,000	46.1	3,600,000	44.2
300,000	56.8	1,500,000	45.3		
400,000	55.5	1,600,000	44.7		

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_XX_XX_88							
	82_17*	83_18	83_19	83_20	83_21	84_22	85_23	
5,000	97.3	76.3	88.6	84.9	87.3	108.9	151.0	
10,000	100.6	93.9	65.2	75.3	72.2	106.0	113.4	
15,000	91.8	76.4	93.1	96.2	96.0	100.9	150.1	
20,000	107.9	79.0	57.7	59.9	76.5	102.5	120.4	
25,000	119.7	80.6	62.0	59.9	72.2	114.2	94.6	
30,000	89.1	91.0	88.2	95.2	84.3	87.9	95.8	
35,000	100.6	82.1	67.6	65.1	71.7	103.7	100.7	
40,000	95.0	85.3	85.2	71.4	76.0	105.5	93.6	
45,000	94.2	67.2	54.9	63.8	74.2	117.6	106.8	
50,000	87.6	96.8	69.8	68.3	64.9	112.0	91.0	
55,000	94.5						116.1	
60,000	87.5						99.3	
65,000	93.0						106.7	
70,000	109.8						93.5	
75,000	102.9						86.8	
80,000	99.6						88.9	
85,000	106.5						91.4	
90,000	93.9						83.5	
95,000	101.6						96.4	
100,000	91.6						81.6	
105,000							108.4	
110,000							94.4	
115,000							106.0	
120,000							88.5	
125,000							85.2	
130,000							85.9	
135,000							93.0	
140,000							83.6	
145,000							97.3	
150,000							85.4	

* This should be model SMB_**U**_A_82_17_88 (not SMB_**M**_A_82_17_88)

E.3 28 Day Forecasts

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_80_XX_XX							
	14_88	15_88	16_88	14_89	15_89	16_89	14_90	15_90
1,000	83.4	88.4	102.9	93.3	95.2	89.6	82.9	88.0
2,000	84.7	93.3	84.7	97.0	99.1	92.7	69.2	84.3
3,000	84.0	98.0	90.7	97.2	101.7	94.6	63.7	79.4
4,000	83.4	102.2	66.5	92.7	101.0	92.9	65.0	76.9
5,000	69.0	82.2	63.7	94.6	105.5	87.5	51.7	75.0
6,000	74.3	94.8	60.4	98.4	104.5	89.7	62.2	71.5
7,000	62.3	82.6	57.9	95.4	102.2	93.8	52.6	76.6
8,000	84.8	110.6	73.2	90.4	105.1	85.4	51.0	75.3
9,000	63.5	81.3	69.0	91.1	103.6	86.0	61.8	78.9
10,000	68.1	79.5	57.2	88.7	103.8	83.3	57.0	73.5
11,000	61.2	81.6	60.2	88.6	102.9	91.2	51.7	73.4
12,000	61.7	80.9	57.1	92.2	101.2	79.0	52.7	68.9
13,000	53.8	77.2	60.2	92.6	105.1	81.2	52.4	73.5
14,000	61.3	78.3	67.5	81.8	95.4	79.5	59.4	71.7
15,000	65.5	82.6	59.3	84.4	100.8	77.2	68.4	85.3
16,000	66.1	84.0	60.6	86.0	95.0	76.7	50.0	74.0
17,000	56.3	77.2	63.1	82.6	94.6	80.1	50.4	80.9
18,000	62.7	84.8	62.6	82.8	93.2	71.6	49.9	80.6
19,000	73.5	88.4	74.2	84.0	92.7	73.0	51.1	72.8
20,000	87.9	103.4	76.2	76.8	87.9	76.4	54.2	71.2
21,000				79.1	88.0	73.9		
22,000				85.4	90.7	75.5		
23,000				77.7	88.3	73.9		
24,000				77.0	89.5	75.8		
25,000				80.4	90.4	70.5		
26,000				80.8	91.1	68.4		
27,000				79.6	95.5	74.7		
28,000				82.1	91.8	70.8		
29,000				79.7	90.8	76.5		
30,000				75.4	91.9	76.1		

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_80_XX_XX (cont.)							
	14_88	15_88	16_88	14_89	15_89	16_89	14_90	15_90
31,000				79.7	89.5	75.1		
32,000				79.0	94.7	81.0		
33,000				78.8	89.3	71.3		
34,000				77.1	91.6	74.5		
35,000				85.2	90.0	68.4		
36,000				76.0	85.3	75.7		
37,000				83.0	88.6	74.8		
38,000				76.3	89.8	72.5		
39,000				79.9	90.0	75.1		
40,000				77.5	93.4	76.1		

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_80_XX_XX							
	16_90	14_91	15_91	16_91				
1,000	85.8	83.8	100.5	97.3				
2,000	70.0	78.3	95.1	92.9				
3,000	58.7	78.1	92.1	89.4				
4,000	53.9	67.6	89.0	84.9				
5,000	63.1	80.0	88.2	84.1				
6,000	55.1	78.0	87.7	83.1				
7,000	52.6	78.2	88.3	80.1				
8,000	58.7	74.9	86.4	79.0				
9,000	51.2	79.6	85.6	77.4				
10,000	48.3	75.9	86.2	79.0				
11,000	55.8	84.5	85.3	76.2				
12,000	55.4	78.5	86.0	75.2				
13,000	52.5	76.0	84.0	75.1				
14,000	48.3	85.1	86.4	78.2				
15,000	51.7	92.6	86.2	79.0				
16,000	47.9	74.4	84.9	75.5				
17,000	48.2	69.6	85.1	78.5				
18,000	57.3	76.8	80.6	78.8				
19,000	47.2	79.2	82.5	78.8				
20,000	46.1	69.8	84.6	74.7				
21,000	45.5			76.0				
22,000	59.4			73.7				
23,000	52.2			72.9				
24,000	46.7			73.6				
25,000	52.8			73.2				
26,000	44.0			71.6				
27,000	42.0			80.9				
28,000	55.1			76.1				
29,000	43.2			70.9				
30,000	44.3			78.8				

Learn Count	RMSE (EC Units)							
	Model SMB_M_A_80_XX_XX (cont.)							
	16_90	14_91	15_91	16_91				
31,000				77.2				
32,000				74.6				
33,000				75.2				
34,000				75.8				
35,000				79.2				
36,000				74.4				
37,000				74.7				
38,000				76.9				
39,000				75.0				
40,000				71.9				

Bibliography

[1] Abraham, B. (1993) "Multiple Time-series Modelling - Another Look at the Canadian Money and Income Data", *Journal of Forecasting*, Vol.12, pp.449-458.

[2] Abraham, B. and Ledolter, J. (1983) "*Statistical Methods for Forecasting*", John Wiley, New York, New York.

[3] Ackley, D.H., Hinton, G.E., and Sejnowski, T.J. (1985) "A Learning Algorithm for Boltzmann Machines", *Cognitive Science*, Vol.9, pp.147-169.

[4] Adeli, H. and Yeh, L. (1990) "Neural Network Learning in Engineering Design", *Proccedings of the International Neural Network Conference*, Vol.1, Paris, July 9-13, pp.412-415.

[5] Akaike, H. (1973) "Maximum Likelihood Identification of Gaussian Auto-regressive Moving Average Models", *Biometrica*, Vol.66, pp.229-236.

[6] Akaike, H. (1974) "A New Look at the Statistical Model Identification", *IEEE Trans. on Auto Control*, Vol.19, pp.716-723.

[7] Ali, M. (1990) "Intelligent Systems in Aerospace", *Knowledge Engineering*, Rev.5, pp.147-166.

[8] Australian Mineral Development Laboratories (AMDEL) (1983) "*Summary on Effects of Salinity on Municipal and Industrial Water Consumers in Respect of the River Murray, South Australia*", May.

[9] Anderson, J.A. (1972) "A Simple Neural Network Generating an Interactive Memory", *Mathematical Biosciences*, Vol.14, pp.197-220.

[10] Anderson, J.A. (1983) "Cognitive and Psychological Computation with Neural Models", *IEEE Transaction on Systems, Man, and Cybernetics*, Vol. SMC-13(5), September/October, pp.799-815.

[11] Anderson, O.D. (1977) "Time Series Analysis and Forecasting: Another Look at the Box-Jenkins Approach", *The Statistician*, Vol.26, No.4, pp.285-297.

[12] Ansley, C.F. and Kohn, R. (1984) "On the Estimation of ARIMA Models with Missing Values", in *Time Series Analysis of Irregularly Observed Data*, edited by E. Parzen, Springer-Verlag, New York.

[13] Aoki M. (1987) "*State Space Modeling of Time Series*", Springer-Verlag, 314p.

[14] Atlas, L., Cole, R., Muthusamy, Y., Lippmann, A., Connor, J., Park, D., El-Sharkaw, M. and Marks, R.J. (1990) "A Performance Comparison of Trained Multi-Layer Perceptrons and Trained Classification Trees", *Proceedings of the IEEE*, Vol.78, No.10, October, pp.1614-1619.

[15] Baalsrud, K. (1982) "The Habilitation of Norway's Largest Lake", *Progress in Water Technology*, Vol.14, pp.21-30.

[16] Baracos, P.C., Hipel, K.W. and McLeod, A.I. (1981) "Modelling Hydrologic Time Series from the Arctic", *Water Resources Bulletin*, Vol.17, No.3, pp.414-422.

[17] Baum, E.B. and Haussler, D. (1989) "What Size Net Gives Valid Generalization?", *Neural Computation*, Vol.1, pp.151-160.

[18] Behndorf, J. and Recknagel, F. (1982) "Problems of Application of the Ecological Model SALMO to Lakes and Reservoirs Having Various Trophic States", *Ecological Modelling*, Vol.17, pp.129-145.

[19] Bergman, M.J. and Delleur, J.W. (1985a) "Kalman Filter Estimation and Prediction of Daily Stream Flows: I. Review, Algorithm, and Simulation Experiments", *Water Resources Bulletin*, Vol.21, No.5, pp.827-8832.

[20] Bergman, M.J. and Delleur, J.W. (1985b) "Kalman Filter Estimation and Prediction of Daily Stream Flows: II. Application to the Potomac River", *Water Resources Bulletin*, Vol.21, No.5, pp.815-825.

[21] Bishop, C.M. (1990) "Curvature-Driven Smoothing in Backpropagation Neural Networks", *Proceedings of the International Neural Network Conference*, Vol.1, Paris, July 9-13, pp.749-752.

[22] Blyth, S. (1980) "Palm Island Mystery Disease", *Medical Journal of Australia*, Vol.2, pp.40-42.

[23] Bowermann, B.L. and O'Connell, R.T. (1979) "*Time Series and Forecasting*", Duxbury Press, North Scituate, Massachusetts, 481p.

[24] Bowmer, K.H., Padovan, A., Oliver, R.L., Korth, W. and Ganf, G.G. (1992) "Physiology of Geosmin Production by *Anabaena circinalis* from the Murrumbidgee River, Australia", *Water Science and Technology*, Vol.25, pp.259-268.

[25] Box, G.E.P. and Cox, D.R. (1964) "An Analysis of Transformations", *J. Roy. Statistic. Soc.*, B, Vol.26, pp.211-243.

[26] Box, G.E.P. and Jenkins, G.M. (1976) "*Time Series Analysis, Forecasting and Control*", Holden-Day Inc., San Francisco, California, 553p.

[27] Box, G.E.P. and Pierce, D.A. (1970) "Distribution of Residual Auto-correlations in Autoregressive-Integrated Moving Average Time-series Models", *J. Amer. Statist. Ass.*", Vol.65, pp.1509-1526.

[28] Brillinger, D.R. (1985) "Fourier Inference: Some Methods for the Analysis of Array and NonGaussian Series Data", *Water Resources Bulletin*, Vol.21, No.5, pp.743-756.

[29] Brockwell, P.J. and Davis, R.A. (1987) "*Time Series: Theory and Methods*", Springer Verlag, New York, 519p.

[30] Broomhead, D.S. and Lowe, D. (1988) "Radial Basis Functions, Multi-Variable Interpolation and Adaptive Networks", *Royal Signals and Radar Establishment Memo. 4148*, Worcestire, United Kingdom.

[31] Burch, M.D. (1993) "Cyanobacteria and Water Quality: New Problems and Challenges", *Drinking Water Quality: Technical and Management Issues for the 1990's*, Seminar, Adelaide, October 12, pp.18-24.

[32] Burch, M.D. (1995), Australian Centre for Water Quality Research, Personal Communication.

[33] Buringh, P. (1979) "Food Production Potential of the World", in: *The World Food Problem; Concesus and Conflict*, Radhe Sinha, Ed, Pergamom Press, New York, pp.477-485.

[34] Burke, L.I. and Ignizio, J.P. (1992) "Neural Networks and Operations Research: An Overview", *Computer and Operations Research*, Vol.19, No.3/4, April/May, pp.179-189.

[35] Camacho, F., McLeod, A.I. and Hipel, K.W. (1983) "The Use and Abuse of Multivariate Time Series in Hydrology", *Technical Report 83-12*, Department of Statistics and Actuarial Sciences, The University of Western Ontario, Canada.

[36] Camacho, F., McLeod, A.I. and Hipel, K.W. (1985) "Contemporaneous Autoregressive-Moving Average (CARMA) Modelling in Water Resources", *Water Resources Bulletin*, Vol.21, No.4, pp.709-720.

[37] Canu, S., Sorbal, R. and Lengelle, R. (1990) "Formal Neural Network as an Adaptive Model for Water Demand", *Proceedings of the International Neural Network Conference*, Vol.1, Paris, July 9-13, pp.131-136.

[38] Carmichael, W.W. (1994) "An Overview of Toxic Cyanobacteria Research in the United States", *Toxic Cyanobacteria - A Global Perspective*, Seminar, Adelaide, South Australia, March 28, pp.24-29.

[39] Chakraborty, K., Mehrotra, K., Mohan, C.K. and Ranka, S. (1992) "Forecasting the Behaviour of Multivariate Time Series Using Neural Networks", *Neural Networks*, Vol.5, pp.961-970.

[40] Chatfield, C. (1975) "*The Analysis of Time Series: Theory and Practice*", Chapman and Hall Ltd, 263p.

[41] Cheung, R.K.M., Lustig, I. and Kornhauser, A.L. (1990) "Relative Effectiveness of Training Set Patterns for Back Propagation", *Proceedings of the IJCNN International Joint Conference on Neural Networks*, San Diego, California, June 17-21, pp.I673-I678.

[42] Christensen R. (1991) "*Linear Models for Multivariate, Time Series, and Spatial Data*", Springer-Verlag, 313 p.

[43] Chung, F.L. and Lee, T. (1992) "A Node Pruning Algorithm for Backpropagation Networks", *International Journal of Neural Systems*, Vol.3, pp.301-314.

[44] Codd, G.A., Bell, S.G. and Brooks, W.P. (1989) "Cyanobacterial Toxins in Water", *Water Science and Technology*, Vol.21, pp.1-13.

[45] Cooper, D.M. and Wood, E.F. (1982a) "Identification of Multivariate Time Series and Multivariate Input-Output Models", *Water Resources Research*, Vo.18, No.4, pp.937-946.

[46] Cooper, D.M. and Wood, E.F. (1982b) "Parameter Estimation of Multiple Input-Output Time Series Models: Application to Rainfall-Runoff Processes", *Water Resources Research*, Vo.18, No.5, pp.1352-1364.

[47] Cottrell, G.W. and Fleming, M. (1990) "Face Recognition Using Unsupervised Feature Extraction" *Proceedings of the International Neural Networks Conference*, Vol.1, Paris, France, pp.322-325.

[48] Creagh, C. (1992) "What Can Be Done About Toxic Algal Blooms?", *Ecos*, Vol.72, pp.14-19.

[49] Croome, R. (1980) "Turbidity, Colour, Nutrients, and Plankton of the River Murray", *A Report to the River Murray Commission Water Quality Committee*, April.

[50] Cryer, J. D. (1986) "*Time Series Analysis*", Duxbury Press, Boston, 286p.

[51] Cybenko, G. (1988) "Continuous Valued Neural Networks with Two Hidden Layers are Sufficient", *Technical Report*, Computer Science Department, Tufts University, Medford, MA.

[52] Dagli, C.H. and Lammers, S. (1989) "Possible Applications of Neural Networks in Manufacturing", *Proceedings of the International Joint Conference on Neural Networks*, Vol.2, Washington, DC, p.605.

[53] Dandy, G.C. and Crawley, P.D. (1992) "Optimisation of Multiple Reservoir Systems including Salinity Effects", *Water Resources Research*, 28(4), pp.979-990.

[54] Daniell, T.M. (1991) "Neural Networks - Applications in Hydrology and Water Resources Engineering", *International Hydrology and Water Resources Symposium*, Perth, 2-4 October, pp.797-802.

[55] Daniell, T.M. and Wundke, A.D. (1993) "Neural Networks - Assisting in Water Quality Modelling", *Watercomp*, Melbourne, Australia, March 30 - April 1, pp.51-57.

[56] Davis, D.L., Gillenwater, E.L. and Johnson, J.D. (1990) "An Artificial Neural Systems Framework for Delivery Truck Scheduling", *Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences*, Vol.3, Kailua-Kona, HI, USA, pp.327-333, IEEE Computer Society Press, Los Alamitos, California, USA.

[57] Davis, J.C. (1973) *"Statistics and Data Analysis in Geology"*, John Wiley & Sons, 637p.

[58] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977) "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society, Series B*, Vol.39, pp.1-38.

[59] DeSieno, D. (1988) "Adding a Conscience to Competitive Learning", *Proceedings of the Second Annual IEEE International Conference on Neural Networks*, Vol.1.

[60] De Silets, L., Golden, B., Wang, Q. and Kumar, R. (1992) "Predicting Salinity in the Chesapeake Bay Using Backpropagation", *Computer and Operations Research*, Vol.19, No.3/4, April/May, pp.277-285.

[61] Dillenberg, H.O. and Dehnel, M.K. (1959) "Toxic Water Bloom in Saskatchewan", *Canadian Medical Association Journal of the Canadian Medical Association*, Vol.83, pp.1151-1154.

[62] Dillon, P.J and Rigler, F.H. (1974) "The Phosphorus-Chlorophyll Relationship in Lakes", *Limnology and Oceanography*, Vol.19, pp.767-773.

[63] Donati, C., Drikas, M., Hayes, R. and Newcombe, G. (1993) "Adsorption of Microcystin-LR by Powdered Activated Carbon", *Water*, Vol.20, No.3, pp.25-28.

[64] Donati, C., Drikas, M., Hayes, R. and Newcombe, G. (1994) "Activated Carbon in Drinking Water Treatment: Part 2, Adsorption of Nodularin", *Poster Presentation at the 8th International Conference, Surface and Colloid Science*, Adelaide, February.

[65] Drikas, M. (1994) "Removal of Cyanobacterial Toxins by Water Treatment Processes", *Toxic Cyanobacteria - A Global Perspective*, Adelaide, South Australia, March 28, pp.30-44.

[66] Dwyer Leslie Pty Ltd. (1984) "River Murray Water Quality Management Study", *Working Papers E and F, The Economic Impacts of Salinised Water Supply*, Maunsell and Partners, Melbourne, Australia, November.

[67] Egashira, K., Ito, K. and Yoshiy, Y. (1992) "Removal of Musty Odour Compound in Drinking Water by Biological Filter", *Water Science and Technology*, Vol.25, pp.307-314.

[68] Engineering and Water Supply Department (EWS) (1987) "*Woolpunda Groundwater Interception Scheme - Draft Environmental Impact Statement*".

[69] Engineering and Water Supply Department (EWS) (1989) "Controlling the River Murray in South Australia", *Information Bulletin No. 17*, October.

[70] Engineering and Water Supply Department (EWS) (1990) "Salinity in the River Murray", *Information Bulletin No. 15*, April.

[71] Ericsson, N.R. and Hendry, D.F. (1985) "Conditional Econometric Modelling: An Application to New House Prices in the United Kingdom", in *A Celebration of Statistics*, A.C. Atkinson and S.E. Fienberg, Eds, Springer Verlag, New York.

[72] Ezhov, A. and Sala, E.D. (1994) "Fractal Neural Classification System and its Application to Music Symbols Recognition", *Neural Network World*, Vol.4, No.2, pp.121-131.

[73] Fahlman, S.E. and Lebiere, C. (1990) "The Cascade-Correlation Learning Architecture", in *Advances in Neural Information Processing Systems 2*, D. Touretzky, Ed., Morgan Kaufmann, San Mateo, pp.524-532.

[74] Falconer, I.R. (1991) "Tumor Promotion and Liver Injury Caused by Oral Consumption of Cyanobacteria", *Environ. Toxicol. Water Qual.*, Vol.6, pp.177-184.

[75] Falconer, I.R. (1994) "Health Implications of Cyanobacterial (Blue-green Algal) Toxins", *Toxic Cyanobacteria - A Global Perspective*, Seminar, Adelaide, South Australia, March 28, pp.2-6.

[76] Falconer, I.R., Jackson, A.R.B., Langley, J. and Runnegar, M.T. (1981) "Liver Pathology in Mice Poisoning by the Blue-green Alga *Microcystis aeruginosa*", *Australian Journal of Biological Sciences*, Vol.34, pp.179-187.

[77] Falconer, I.R., Beresford, A.M., and Runnegar, M.T.C. (1983a) "Evidence of Liver Damage by Toxin from a Bloom of the Cyanobacterium *Microcystis aeruginosa*", *Medical Journal of Australia*, Vol.1, pp.511-514.

[78] Falconer, I.R., Runnegar, M.T.C. and Huyn, V.L. (1983b) "Effectiveness of Activated Carbon in the Removal of Algal Toxin from Potable Water Supplies: A Pilot Plant Investigation", *Proc. 12th AWWA Federal Convention*, Sydney, pp.26-1 - 26-8.

[79] Falconer, I.R., Smith, J.V., Jackson, A.R.B., Jones, A. and Runnegar, M.T.C. (1988) "Oral Toxicity of a Bloom of the Cyanobacterium *Microcystis*

aeruginosa Administered to Mice Over Periods up to 1 Year", *J. Toxicol. Environ. Health*, Vol.24, pp.291-305.

[80] Falconer, I.R., Runnegar, M.T.C., Buckley, T., Huyn, V.L. and Bradshaw, P. (1989) "Using Activated Carbon to Remove Toxicity from Drinking Water Containing Cyanobacterial Blooms", *Journal of the Australian Water and Wastewater Association*, Vol.81, No.2, pp.102-105.

[81] Falconer, I.R., Dornbusch, M., Moran, G., and Yeung, S.K. (1992) "Effect of the Cyanobacterial (Blue-green Algal) Toxins from *Microcystis aeruginosa* on Isolated Enterocytes from the Chicken Small Intestine", *Toxicon*, Vol.30, pp.790-793.

[82] Filliben, J.J. (1975) "The Probability Plot Correlation Coefficient Test for Normality", *Technometrics*, Vol.17, No.1, pp.111-117.

[83] Fleming N.S. (1994) "Forecasting Water Consumption in the Northern Adelaide Plains, SA, Using Artificial Neural Networks, Regression and Time Series Models", *Research Report No. R115*, Department of Civil and Environmental Engineering, The University of Adelaide, Adelaide, March, 45 p.

[84] Francis, G. (1878) "Poisonous Australian Lake", *Nature*, Vol.18, pp.11-12.

[85] French, M. and Recknagel, F. (1994) "Modelling of Algal Blooms in Freshwaters Using Artificial Neural Networks", in *Computer Techniques in Environmental Studies V, Vol.II: Environmental Systems*, P. Zanetti, Ed, Computational Mechanics Publications, Southampton, Boston, pp.87-94.

[86] Fukushima, K. (1988) "Neocognition: A Hierarchical Neural Network Capable of Pattern Recognition", *Neural Networks*, Vol.1, pp.119-130.

[87] Galey, F.D., Beasley, V.R., Carmichael, W.W., Kleppe, G., Hooser, S.B. and Haschek, W.M. (1987) "Blue-green Algae (*Mycrocystis aeruginosa*) Hepatotoxicosis in Dairy Cows", *American Journal of Veterinary Research*, Vol.48, pp.1415-1420.

[88] Gaspin, C. (1989) "Mission Scheduling", *Telematics and Informatics*, Vol.6, pp.159-169.

[89] Ghassemi, F., Jakeman, A.J. and Nix, H.A. (1991) "Human Induced Salinisation and the Use of Quantitative Methods", *Environment International*, Vol.17, pp.581-594.

[90] Gilmore, J.F. and Czuchry Jr., A.J. (1990) "An Application of the Neocognitron in Target Recognition" *Proceedings of the International Neural Networks Conference*, Vol.1, Paris, France, pp.15-18.

[91] Glatz, A (1994) "Water Quality: Inorganic Aspects", *Lecture Notes*, Water Quality Fundamentals and Processes, The University of South Australia.

[92] Goldberg, D.E. (1989) "*Genetic Algorithms in Search, Optimization and Machine Learning*", Addison-Wesley Publishing Company, Inc., 412p.

[93] Gosh, J. and Bovik, A.C. (1991) "Neural Networks for Textured Image Processing", in *Artificial Neural Networks and Statistical Pattern Recognition - Old and New Connections*, pp.133-154, I.K. Sethi and A.K. Jain, Eds, Elsevier Science Publishers B.V., Amsterdam.

[94] Granger, C.W.J. (1966) "The Typical Shape of an Econometric Variable", *Econometrica*, Vol.34, pp.150-161.

[95] Granger, C.W.J. (1969) "Investigating Causal Relations by Econometric Models and Cross-Spectral Methods", *Econometrica*, Vol.37, No.3, pp.424-438.

[96] Granger, C.W.J. (1980) "*Forecasting in Business and Economics*", Academic Press, New York, New York.

[97] Granger, C.W.J. and Newbold, P. (1977) "*Forecasting Economic Time Series*", Academic Press, New York, New York.

[98] Gray, H.L., Kelley, G. and McIntire, D. (1978) "A New Approach to ARMA Modelling", *Comm. Statist.*, B, Vol.7, pp.1-78.

[99] Grossberg, S. (1976) "Adaptive Pattern Classification and Universal Recording, 1: Parallel Development and Coding of Neural Feature Detectors", *Biological Cybernetics*, Vol.23, pp.121-134.

[100] Grossberg, S. (1978) "A Theory of Human Memory: Self-Organization and Performance of Sensory-Motor Codes, Maps and Plans", in *Progress in Theoretical Biology*, Vol.5, R. Rosen and F. Sness, Eds., Academic, New York, NY.

[101] Grossberg, S. and Mingolla, E. (1985a) "Neural Dynamics of Form Perception: Boundary Completion, Illusory Figures and Neon Colour Spreading", *Psychological Review*, Vol.92, pp.173-211.

[102] Grossberg, S. and Mingolla, E. (1985b) "Neural Dynamics of Perceptual Grouping: Textures, Boundaries and Emergent Segmentations", *Perception and Psychophysics*, Vol.38, pp.141-171.

[103] Gupta, N.K. and Mehra, R.K. (1974) "Computational Aspects of Maximum Likelihood Estimation and Reduction in Sensitivity Function Calculations", *IEEE Transactions on Auto. Contro*l, AC-19, pp.774-783.

[104] Halff, A.H. and Halff, H.M. (1993) "Predicting Runoff from Rainfall Using Neural Networks", *Proceedings of the Symposium on Engineering Hydrology*, San Francisco, California, July 25-30, pp.760-765.

[105] Haltiner, J.P. and Salas, J.D. (1988) "Short-term Forecasting of Snowmelt Runoff Using ARMAX Models", *Water Resources Bulletin*, Vo.24, No5, pp.1083-1089.

[106] Hancox, P.J., Mills, J., Reid, J. (1990) "*Artificial Intelligence / Expert Systems*", Mansell, London, pp.6-7.

[107] Hannan, E.J. and Deistler, M. (1988) "*The Statistical Theory of Linear Systems*", John Wiley, New York.

[108] Harada, K. (1994) "Strategy for Trace Analysis of Microcystins in Complicated Matrix", *Toxic Cyanobacteria - A Global Perspective*, Seminar, Adelaide, South Australia, March 28, pp.49-51.

[109] Harp, A.N., Samad, T, and Guha, A. (1989) "The Genetic Synthesis of Neural Networks", *Honeywell Technical Report CSDD-89-I4852-2*, June.

[110] Harrison, P.J. and Stevens, C.F. (1971) "A Bayesian Approach to Short-term Forecasting", *Operational Research Quarterly*, Vol.22, No.4, pp.341-362.

[111] Harrison, P.J. and Stevens, C.F. (1976) "Bayesian Forecasting", *Journal of the Royal Statistical Society, Series B*, Vol.38, pp.205-247.

[112] Harvey A.C. (1981) "*Time Series Models*", Philip Allan Publishers Limited, Oxford, 225 p.

[113] Harvey, A.C. (1984) "A Unified View of Statistical Forecasting Procedures", *Journal of Forecasting*, Vol.3, pp.245-275.

[114] Harvey A.C. (1989) "*Forecasting, Structural Time Series Models and the Kalman Filter*", Cambridge University Press, 547 p.

[115] Harvey, A.C. and Peters, S. (1990) "Estimation Procedures for Structural Time Series Models", *Journal of Forecasting*, Vol.9, pp.89-108.

[116] Haugh, L.D. and Box, G.E.P. (1977) "Identification of Dynamic Regression (Distributed Lag) Models Connecting Two Time Series", *Journal of the American Statistical Association*, Vol.72, No.397, pp.121-130.

[117] Hawkins, P.R., Runnegar, M.T.C. and Falconer, I.R. (1985) "Severe Hepatotoxicity Caused by the Tropical Cyanobacterium (Blue-green Alga) *Cylindrospermopsis raciborskii* (Woloszynska) Seenaya and Subba Raju Isolated from a Domestic Water Supply Reservoir", *Applied Environmental Microbiology*, Vol.50, pp.1292-1295.

[118] Hebb, D.O. (1949) "*The Organization of Behaviour*", Wiley, New York.

[119] Hecht-Nielsen, R. (1987a) "Counterpropagation Networks", *Applied Optics*, Vol.26(23), pp.4979-4984.

[120] Hecht-Nielsen, R. (1987b) "Kolmogorov's Mapping Neural Network Existence Theorem", *Proceedings of the First IEEE International Joint Conference on Neural Networks,* San Diego, California, June 21-24, pp.III11-III14.

[121] Hecht-Nielsen, R. (1988) "Applications of Counterpropagation Networks", *Neural Networks*, Vol.1, pp.131-139.

[122] Hecht-Nielsen, R. (1988) "Neurocomputing: Picking the Human Brain", *IEEE Spectrum*, Vol. 25(3), March, pp.36-41.

[123] Hecht-Nielsen, R. (1990) "*Neurocomputing*", Addison-Wesley Publishing Company, 433 p.

[124] Hendry, D.F. and Richard, J.-F. (1983) "The Econometric Analysis of Economic Time Series", *International Statistical Review*, Vol.51, pp.111-164.

[125] Hertz, J.A., Krogh,A. and Palmer, R.G. (1991) "*Introduction to the Theory of Neural Computation*", Addison-Wesley Publishing Company, Redwood City, California.

[126] Himberg, K., Keijola, A.M., Hiisvirta, L., Pyysalo, H. and Sivonen, K. (1989) "The Effect of Water Treatment Processes on the Removal of Hepatotoxins from *Microcystis* and *Oscillatoria* Cyanobacteria: A Laboratory Study", *Water Research*, Vol.23, No.8, pp.979-984.

[127] Hinton, G.E. and McClelland, J.L. (1988) "Learning Representations by Recirculation", *Proceedings of IEEE Conference on Neural Information Processing Systems*, November.

[128] Hipel, K.W. (1985) "Time Series Analysis in Perspective", *Water Resources Bulletin*, Vol.21, No.4, pp.609-624.

[129] Hipel, K.W., McLeod, A.I. and Lennox, W.C. (1977a) "Advances in Box-Jenkins Modelling 1. - Model Construction", *Water Resources Research*, Vol.13, No.3, June, pp.567-575.

[130] Hipel, K.W., McLeod, A.I. and McBean, E.A. (1977b) "Stochastic Modelling of the Effects of Reservoir Operation", *Journal of Hydrology*, Vol.32, pp.97-113.

[131] Hishida, Y., Ashitani, K. and Fujiwara, K. (1988) "Occurrence of Musty Odor in the Yodo River", *Water Science and Technology*, Vol.20, pp.193-196.

[132] Hodgkin, E.P., Birch, P.B., Black, R.E. and Humphries, R.B. (1980) "The Peel-Harvey Estuarine System Study (1976-1980): A Report to the Estuarine and

Environment Committee", *Report No. 9*, Department of Conservation & Environment, December.

[133] Hoffmann, J.R.H. (1976) "Removal of *Microcystis* Toxins in Water Purification Processes, *Water SA*, Vol.2, pp.58-60.

[134] Honkanen, R.E., Zwiller, J., Moore, R.E., Daily, S.L., Khatra, B.S., Dukelow, M. and Boynton, A.L. (1990) "Characterization of Microcystin-LR, a Potent Inhibitor of Type 1 and Type 2A Protein Phosphatases", *Journal of Biological Chemistry*, Vol.265, pp.19401-19404.

[135] Hooser, S.B., Beasley, V.R., Lovell, R.A., Carmichael, W.W. and Haschek, W.M. (1989) "Toxicity of Microcystin-LR, a Cyclic Heptapeptide Hepatotoxin from *Microcystis aeruginosa*, to Rats and Mice", *Veterinary Pathology*, Vol.26, pp.246-252.

[136] Hooser, S.B., Beasley, V.R., Basgall, E.J., Carmichael, W.W. and Haschek, W.M. (1990) "Microcystin-LR Induced Ultrastructural Changes in Rats", *Veterinary Pathology*, Vol.27, pp.9-15.

[137] Hopfield, J.J. (1982) "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proceedings of the Natural Academy of Science USA*, Vol.79, pp.2254-2558.

[138] Hopfield, J.J. (1984) "Neurons with Graded Response have Collective Computational Properties Like those of Two-State Neurons", *Proceedings of the Natural Academy of Science USA*, Vol.81, pp.3088-3092.

[139] Hopfield, J.J. and Tank, D.W. (1985) "Neural Computations of Decisions in Optimization Problems", *Biological Cybernetics*, Vol.52, pp.141-152.

[140] Hrudey, S.E., Kenefick, S.L., Best, N., Gillespie, T., Kotak, B.G., Prepas, E.E. and Peterson, H.G. (1993) "Liver Toxins and Odour Agents in Cyanobacterial Blooms in Alberta Water Supplies", in: *Disinfection Dilemma: Microbiological Control versus By-Products, Proceedings of the 5th National Conference on Drinking Water*, American Water Works Association, pp.383-390.

[141] Hrudey, S.E., Kenefick, S.L., Lambert, T.W., Kotak, B.G., Prepas, E.E. and Holmes, C.F.B. (1994a) "Sources of Uncertainty in Assessing the Health Risk of

Cyanobacterial Blooms in Drinking Water", *First International Symposium on Detection Methods for Cyanobacterial Toxins*, Codd, G., Ed, Royal Society of Chemistry, Cambridge, U.K., in press.

[142] Hrudey, S.E., Lambert, T.W. and Kenefick, S.L. (1994b) "Health Risk Assessment of Microcystins in Drinking Water Supplies", *Toxic Cyanobacteria - A Global Perspective*, Seminar, Adelaide, South Australia, March 28, pp.7-16.

[143] Hrycej, T. (1990) "A Modular Architecture for Efficient Learning", *Proceedings of the IJCNN International Joint Conference on Neural Networks*", San Diego, California, June 17-21, pp.I557-I562.

[144] Hubick, K.T. (1992) "*Artificial Neural Networks in Australia*", Department of Industry, Technology and Commerce, Commonwealth of Australia, Canberra, 131p.

[145] Hush, D.R. and Horne, B.G. (1993) "Progress in Supervised Neural Networks", *IEEE ASSP Magazine*, January, pp.8-37.

[146] Ignizio Burke, L. (1991) "Introduction to Artificial Neural Systems for Pattern Recognition", *Computer and Operations Research*, Vol.18, No.2, pp.211-220.

[147] IMSL, Inc. (1991), IMSL STAT/LIBRARY *User's Manual*, Version 2.0, IMSL, Houston.

[148] Irvine, K.N and Eberhardt, A.J. (1992) "Multiplicative, Seasonal ARIMA Models for Lake Erie and Lake Ontario Water Levels", *Water Resources Bulletin*, Vol.28, No.2, pp.385-396.

[149] Jackson, A.R.B., McInnes, A., Falconer, I.R. and Runnegar, M.T.C. (1984) "Clinical and Pathological Changes in Sheep Experimentally Poisoned by the Blue-green Algae *Microcystis aeruginosa*", *Veterinary Pathology*, Vol.21, pp.102-113.

[150] Jacobs, R.A. (1988) "Increased Rates of Convergence Through Learning Rate Adaption", *Neural Networks*, Vol.1, pp.295-307.

[151] Jakeman, A.J., Thomas, G.A. and Dietrich, C.R. (1986) "Water Resources Management in the River Murray: Models of Salinity Travel Time and Accession and their Application", *CRES Working Paper 1986/21.*

[152] James, A. (1984) "*Water Quality Modelling*", John Wiley and Sons, 233 p.

[153] James, H. and Fawell, J.K. (1991) "Detection and Removal of Cyanobacterial Toxins from Freshwaters", *Foundation of Water Research Report FR 0211.*

[154] Johnston, M.D. and Adorf, H.-M. (1992) "Scheduling With Neural Networks - The Case of the Hubble Space Telescope", *Computer and Operations Research*, Vol.19, No.3/4, April/May, pp.209-240.

[155] Jones, G.J. and Orr, P.T. (1994) "*In situ* Release and Degradation of Microcystin Following Algicide Treatment of a *Microcystis aeruginosa* Bloom in a Recreational Lake, as Determined by HPLC and Protein Phosphatase Inhibition Assay", (Submitted to *Water Research*)

[156] Jones, G., Burch, M., Falconer, I. and Craig, K. (1993) "Cyanobacterial Toxicity", *Technical Advisory Group Report, Murray-Darling Basin Commission, Algal Managament Strategy*, MDBC, Canberra, pp.17-32.

[157] Jones, J.R. and Bachmann, R.W. (1976) "Prediction of Phosphorus and Chlorophyll Levels in Lakes", *J. Water Pollut. Control Fed.*, Vol.48, pp.2176-2182.

[158] Jones, R.H. (1980) "Maximum Likelihood Fitting of ARMA Models to Time Series with Missing Observations", *Technometrics*, Vol.22, pp.389-396.

[159] Jones, R.D., Lee, Y.L., Barnes, C.W., Flake, G.W., Lee, K., Lewis, P.S. and Qian, S. (1990) "Function Approximation and Time Series Prediction with Neural Networks", *Proceedings of the IJCNN International Joint Conference on Neural Networks*", San Diego, California, June 17-21, pp.I649-I665.

[160] Jones, W.P. and Hoskins J. (1987) "Back-Propagation: A Generalized Delta Rule", *Byte Magazine*, October, pp.155-162.

[161] Jorgensen, S.E. (1976) "A Eutrophication Model for a Lake", *Ecological Modelling*, Vol.2, pp.147-162.

[162] Josin, G. (1987) "Neural-Network Heuristics", *Byte Magazine*, October, pp.183-192.

[163] Kalman, R.E. (1960) "A New Approach to Linear Filtering and Prediction", *Journal of Basic Engineering*, Vol.82, pp.34-35.

[164] Kalman R.E. and Bucy R.S. (1961) "New Results in Linear Filtering and Prediction Theory", *Journal of Basic Engineering*, Vol.83, pp.95-108.

[165] Kamijo, K. and Tanigawa, T. (1990) "Stock Price Pattern Recognition - a Recurrent Neural Network Approach", *Proceedings of the IJCNN International Joint Conference on Neural Networks*, San Diego, California, June 17-21, pp.I215 - I221.

[166] Karunanithi, N., Grenney, W.J., Whitley, D. and Bovee, K. (1994) "Neural Networks for River Flow Prediction", *Journal of Computing in Civil Engineering*, Vol.1, No.2, April, pp.201-220.

[167] Keijola, A.M., Himberg, K., Esala, A.L., Sivonen, K. and Hiisvirta, L. (1988) "Removal of Cyanobacterial Toxins in Water Treatment Processes: Laboratory and Pilot-scale Experiments", *Toxic. Assess.*, Vol.3, pp.643-656.

[168] Keleti, G., Sykora, J.L., Maiolie, L.A., Doerfler, D.L. and Campbell, I.M. (1980) "Isolation and Characterisation of Endotoxin from Cyanobacteria (Blue-green Algae)", in: *The Water Environment*, Carmichael, W.W., Ed, Plenum Press, New York and London, pp.447-464.

[169] Kendall, M.G. (1973) "*Time Series*", Griffin, London.

[170] Kenefick, S.L., Hrudey, S.E., Prepas, E.E., Motkosky, N. and Peterson, H.G. (1992) "Odorous Substances and Cyanobacterial Toxins in Prairie Drinking Water Sources", *Water Science and Technology*, Vol.25, No.2, pp.147-154.

[171] Kenefick, S.L., Hrudey, S.E., Peterson, H.G. and Prepas, E.E. (1993) "Toxin Release from *Microcystis aeruginosa* After Chemical Treatment", *Water Science and Technology*, Vol.27, No.3/4, pp.433-440.

[172] Khotanzad, A. and Lu, J.-H. (1991) "Shape and Texture Recognition by a Neural Network", in *Artificial Neural Networks and Statistical Pattern Recognition - Old and New Connections*, pp.109-131, I.K. Sethi and A.K. Jain, Eds, Elsevier Science Publishers B.V., Amsterdam.

[173] Kimoto, T. and Asakawa, K. (1990) "Stock Market Prediction System with Modular Neural Networks", *Proceedings of the IJCNN International Joint Conference on Neural Networks*, San Diego, California, June 17-21, pp.I1-I6.

[174] Kiviranta, J., Namikoshi, M., Sivonen, K., Evans, W.R., Carmichael, W.W. and Rinehart, K.L. (1992) "Structure Determination and Toxicity of a New Microcystin from *Microcystis aeruginosa* Strain 205", *Toxicon*, Vol.30, pp.1093-1098.

[175] Kobayashi, Y. and Nonaka, H. (1990) "Application of Neural Networks to Schedule Integration in Plant Engineering", *Proceedings of the International Neural Networks Conference*, Vol.1, Paris, France, pp.287-290.

[176] Koch, B., Gramith, J.T., Dale, M.S. and Ferguson, D.W. (1992) "Control of 2-Methylisoborneol and Geosmin by Ozone and PEROXONE: a Pilot Study", *Water Science and Technology*, Vol.25, pp.291-298.

[177] Koch, R.W. and Smillie, G.M. (1986) "Bias in Hydrologic Prediction Using Log-Transformed Regression Models", *Water Resources Bulletin*, Vol.22, No.5, pp.717-723.

[178] Kohonen, T. (1982) "Self-Organized Formation of Topologically Correct Feature Maps", *Biological Cybernetics*, Vol.43, pp.59-69.

[179] Kohonen, T. et.al. (1988) "Statistical Pattern Recognition with Neural Networks: Benchmark Studies", *Proceedings of the Second Annual IEEE International Conference on Neural Networks*, Vol.1.

[180] Kolmogorov, A.N. (1957) "On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition" [in Russian], *Dokl. Kaad. Nauk USSR*, Vol.114, pp.953-956.

[181] Kosko, B. (1987) "Bidirectional Associative Memories", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.18(1), pp.49-60.

[182] Kovda, V.A. (1983) "Loss of Productive Land Due to Salinisation", *AMBIO 12*, pp.91-93.

[183] Kreyszig, E. (1983) "*Advanced Engineering Mathematics*", Fith Edition, John Wiley and Sons, 988p.

[184] Krovvidy, S. and Wee, W.G. (1990) "A Knowledge Based Neural Network Approach for Waste Water Treatment System", *Proceedings of the IJCNN International Joint Conference on Neural Networks*, San Diego, California, June 17-21, pp.I327-I332.

[185] Kudrycki, T.P. (1988) "Neural Network Implementation of a Medical Diagnosis Expert System", *Master's Thesis*, College of Engineering, University of Cincinnati.

[186] Kuiper-Goodman, T., Gupta, S., Combley, H. and Thomas, B.H. (1994) "Microcystins in Drinking Water: Risk Assessment and Derivation of a Possible Guidance Value for Drinking Water", *Toxic Cyanobacteria - A Global Perspective*, Adelaide, Seminar, South Australia, March 28, pp.17-23.

[187] Lachtermacher, G. (1992) "Backpropagation in Time Series Analysis", *PhD Thesis*, University of Waterloo, Ontario, Canada.

[188] Lachtermacher, G. and Fuller, J.D. (1994) "Backpropagation in Hydrological Time Series Forecasting", in *Stochastic and Statistical Methods in Hydrology and Environmental Engineering, Vol.3, Time Series Analysis in Hydrology and Environmental Engineering*, pp.229-242, K.W. Hipel, A.I. McLeod, U.S. Panu and V.P. Singh, Eds, Kluwer Academic Publishers, Dordrecht, Netherlands.

[189] Lampinen, J. and Oja, E. (1990) "Distortion Tolerant Feature Extraction with Gabor Functions and Topological Coding" in *Proceedings of the International Neural Networks Conference*, Vol.1, Paris, France, pp.301-304.

[190] Lapedes, A. and Farber, R. (1988) "How Neural Nets Work", in *Neural Information Processing Systems*, pp.442-456, D. Anderson, Ed, American Institute of Physics, New York.

[191] LeCun, Y. (1989) "Generalization and Network Design Strategies", in *Connectionism in Perspective*, pp.143-154, R. Pfeiffer, Z. Schreter and F. Fogelman-Soulie, Eds, North-Holland.

[192] Ledolter, J. (1978) "The Analysis of Multivariate Time Series Applied to Problems in Hydrology", *Journal of Hydrology*, Vol.36, pp.327-352.

[193] Lehtimäki, J., Sivonen, K., Luukkainen, R. and Niemelä, S.I. (1994) "The Effects of Incubation Time, Temperature, Light, Salinity and Phosphorous on Growth and Hepatotoxin Production by *Nodularia* Strains", *Archiv für Hydrobiologie*, in press.

[194] Lewis, P.A.W. (1985) "Some Simple Models for Continuous Variate Time Series", *Water Resources Bulletin*, Vol.21, No.4, pp.635-644.

[195] Lippmann, R.P. (1987) "An Introduction to Computing with Neural Nets", *IEEE ASSP Magazine*, April, pp.4-22.

[196] Lippy, E.C. and Erb, J. (1976) "Gastrointestinal Illness at Sewickley, Pa", *Journal of the American Water Works Association*, Vol.68, pp.606-610.

[197] Ljung, G.M. and Box, G.E.P. (1978) "On a Measure of Lack of Fit in Time Series Models", *Biometrica*, Vol.65, pp.67-72.

[198] Loncelle, J. (1990) "Constraining Multilayer Perceptrons: Application to the Contour Detection" *Proceedings of the International Neural Networks Conference*, Vol.1, Paris, France, pp.23-26.

[199] Looi, C.K. (1992) "Neural Network Methods in Combinatorial Optimization", *Computer and Operations Research*, Vol.19, No.3/4, April/May, pp.191-208.

[200] Luthy, J. (1979) "Epidemic Paralytical Shellfish Poisoning in Western Europe 1976", in: *Toxic Dinoflagellate Blooms*, Taylor, D.L. and Seliger, H.H., Eds, Elsevier/North Holland, New York and London, pp.23-28.

[201] Lütkepohl H. (1991) "*Introduction to Multiple Time Series Analysis*", Springer-Verlag, 531 p.

[202] Luukkainen, R., Sivonen, K., Namikoshi, M., Färdig, M., Rinehart, K.L. and Niemelä, S.I. (1993) "Isolation and Identification of Eight Microcystins from 13 *Oscillatoria agardhii* Strains: Structure of a New Microcystin", *Applied Environmental Microbiology*, Vol.59, pp.2204-2209.

[203] Luukkainen, R., Namikoshi, M., Sivonen, K., Rinehart, K.L. and Niemelä, S.I. (1994) "Isolation and Identification of 12 Microcystins from Four Strains and Two Bloom Samples of *Microcystis* spp.: Structure of new Hepatotoxins", *Toxicon*, Vol.32, pp.133-139.

[204] Mackay, N. and Eastburn, D. Eds. (1990), "*The Murray*", Murray Darling Basin Commission, Canberra, Australia, 363p.

[205] MacKintosh, C., Beattie, K., Klumpp, S., Cohen, P. and Codd, G. (1990) "Cyanobacterial Microcystin-LR is a Potent Inhibitor of Protein Phosphatases 1 and 2A from Both Mammals and Higher Plants", *Fed. Europ. Biochem. Soc.*, Vol.264, pp.187-192.

[206] Maier, H.R. (1995) "A Review of Artificial Neural Networks", *Research Report No. 131*, Department of Civil and Environmental Engineering, The University of Adelaide, August.

[207] Maier, H.R. and Dandy, G.C. (1993) "The Application of Artificial Neural Networks to the Prediction of Salinity", *Research Report No. 101*, Department of Civil and Environmental Engineering, The University of Adelaide, February.

[208] Maren, A., Harston, C. and Pap, R. (1990) "*Handbook of Neural Computing Applications*", Academic Press Inc., San Diego, California, 448 p.

[209] Matalas, N.C. (1967) "Mathematical Assessment of Synthetic Hydrology", *Water Resources Research*, Vol.3, No.4, pp.937-945.

[210] Matyas, J. (1965) "Random Optimization", *Automation and Remote Control*", Vol.26, pp.246-253.

[211] Maunsell and Partners (1979) "*Murray Valley Salinity and Drainage*", Consulting Report, Canberra, Australia.

[212] May, V. (1981) "The Occurrence of Toxic Cyanophyte Blooms in Australia in the Water Environment", in: *Algal Toxins and Health*, Carmichael, W.W., Ed, Plenum Press, New York, pp.127-142.

[213] McCulloch, W.S. and Pitts, W. (1943) "A Logical Calculus of the Ideas Imminent in Nervous Activity", *Bulletin and Mathematical Biophysics*, Vol.5, pp.115-133.

[214] McLeod, A.I., Hipel, K.W. and Lennox, Camacho, F. (1983) "Trend Assessment of Environmental Time Series", *Water Resources Bulletin*, Vol.19, pp.537-547.

[215] Mertzanis, E.C. and Austin, J. (1991) "Linear Quadtrees for Neural Based Position Invariant Pattern Recognition" in *Neural Network Applications*, Proceedings of the Second British Neural Network Society Meeting (NCM91), London, October, 1991, pp.63-100, J.G. Taylor, Ed, Springer Verlag, London.

[216] Minai, A.A. and Williams, R.D. (1990) "Back-Propagation Heuristics: A Study of the Extended Delta-Bar-Delta Algorithm", *Proceedings of the IJCNN International Joint Conference on Neural Networks*, San Diego, California, June 17-21, pp.I595-I600.

[217] Minsky, V.A. and Maren, A. (1989) "Representing the Perceptual Organization of Segmented Images Using Hierarchical Scene Structures", *Journal of Neural Network Computing*, Vol.1, (Winter), pp.14-33.

[218] Minsky, M. and Papert, S. (1969) "*Perceptrons: An Introduction to Computational Geometry*", MIT Press, Cambridge MA.

[219] Mitra, S, Pal, S.K. and Kundu, M.K. (1994) "Fingerprint Classification Using a Fuzzy Multilayer Perceptron", *Neural Computing and Applications*, Vol.2, No.4, pp.227-233.

[220] Mizon, G.E. and Richard, J.-F. (1986) "The Encompassing Principle and its Application to Testing Non-nested Hypotheses", *Econometrica*, Vol.54, pp.657-678.

[221] Moody, J. and Darken, C.J. (1989) "Fast Learning in Networks of Locally-Tuned Processing Units", *Neural Computation*, Vol.1, pp.281-294.

[222] Murray-Darling Basin Commission (MDBC) (1990) "Review of the Trial Flow and Salinity Forecasts for the Mid Murray: November 1989 to January 1990", *MDBC Technical Report 90/2*, February

[223] Murray Darling Basin Commission (MDBC) (1993) "Technical Advisory Group Outcomes" MDBC, Canberra.

[224] Murray-Darling Ministerial Council (1988) "*Draft Salinity and Drainage Strategy*", January.

[225] Murray Darling Basin Ministerial Council (MDBMC) (1993) "Algal Management Strategy for the Murray-Darling Basin (Draft)", MDBC, Canberra.

[226] Namikoshi, M., Rinehart, K.L., Sakai, R., Sivonen, K. and Carmichael, W.W. (1990) "Structures of Three New Cyclic Heptapeptide Hepatoxins Produced by the Cyanobacterium (Blue-green Alga) *Nostoc* sp. strain 152", *Journal of Organic Chemistry*, Vol.55, pp.6135-6139.

[227] Namikoshi, M., Sivonen, K., Evans, W.R., Carmichael, W.W., Rouhiainen, L., Luukkainen, R. and Rinehart, K.L. (1992a) "Structures of Three New Homotyrosine Containing Microcystins and a New Homophenylalanine Variant from *Anabaena* sp. Strain 66", *Chem. Res. Toxicol.*, Vol.5, pp.661-666.

[228] Namikoshi, M., Sivonen, K., Evans, W.R., Carmichael, W.W., Sun, F., Rouhiainen, L., Luukkainen, R. and Rinehart, K.L. (1992b) "Two New L-serine Variants of Microcystins-LR and -RR from *Anabaena* sp. Strains 202A1 and 202A2", *Toxicon*, Vol.30, pp.1457-1464.

[229] Namikoshi, M., Sivonen, K., Evans, W.R., Carmichael, W.W. and Rinehart, K.L. (1992c) "Isolation and Structure Determination of Microcystins from a Cyanobacterial Water Bloom (Finland)", *Toxicon*, Vol.30, pp.1473-1479.

[230] National Rivers Authority (UK) (1990) "Toxic Blue-green Algae", *Water Quality Series*, No.2, Stanley L. Hunt (Printers) Ltd., September

[231] Nelson, C.R. (1973) "*Applied Time Series Analysis for Managerial Forecasting*", Holden-Day, San Francisco, California.

[232] NeuralWare, Inc. (1991) *Neural Computing, NeuralWorks Professional II/Plus and NeuralWorks Explorer*, 360 p.

[233] Ng, C.N. and Young, P.C. (1990) "Recursive Estimation and Forecasting of Non-stationary Time Series", *Journal of Forecasting*, Vol.9, pp.173-204.

[234] Nicholson, B.C., Rositano, J., Humpage, A. and Burch, M. (1993) "Removal of Algal Toxins in Water Treatment Processes", *Proc. AWWA 15th Federal Convention*, Gold Coast, April, pp.327-331.

[235] Nicholson, B.C., Rositano and Burch, M. (1994) "Destruction of Cyanobacterial Peptide Hepatotoxins by Chlorine and Chloramine", *Water Research*, in press

[236] Nishiwaki-Mastushima, Otha, T.R., Nishiwaki, S., Suganuma, M., Kohyama, K., Ishikawa, T., Carmichael, W.W. and Fujiki, H. (1992) "Liver Tumor Promotion by the Cyanobacterial Cyclic Peptide Toxin Microcystin-LR", *J. Cancer Res. Clinical Oncol.*, Vol.118, pp.420-424.

[237] Oja, E., Xu, L. and Kultanen, P. (1990) "Curve Detection by an Extended Self-Organising Map and the Related RHT Method" *Proceedings of the International Neural Networks Conference*, Vol.1, Paris, France, pp.27-30.

[238] Omatu, S., Fukumi, M. and Teranisi, M (1990) "Neural Network Model for Alphabetical Letter Recognition" *Proceedings of the International Neural Networks Conference*, Vol.1, Paris, France, pp.19-22.

[239] Ozaki, T. (1985) "Statistical Identification of Storage Models with Application to Stochastic Hydrology", *Water Resources Bulletin*, Vol.21, No.4, pp.663-675.

[240] Pankratz, A. (1983) "*Forecasting With Univariate Box-Jenkins Models*", John Wiley, New York, New York.

[241] Parzen, E. (1974) "Recent Advances in Time Series Modeling", *IEEE Transactions on Auto Control*, Vol.19, pp.723-730.

[242] Patterson, S. (1993) "The Prediction of Water Salinity Using Artificial Neural Networks", *Final Year Undergraduate Research Project Thesis*, University of Adelaide.

[243] Persson, P.-E. (1992) "A Summary of Problem Areas in Aquatic Off-flavour Research", *Water Science and Technology*, Vol.25, pp.335-340.

[244] Pieronne, P. (1993) "*Report on French-Australian Industrial Research Programme*", ACWQR, July.

[245] Pfeiffer, P. (1994) Operations Support Branch, Engineering and Water Supply Department of South Australia, Personal Communication.

[246] Priestley M.B. (1988) "*Non-linear and Non-stationary Time Series Analysis*", Academic Press Inc., San Diego, California, 235 p.

[247] Radbergh, C.M.I., Bylund, G. and Eriksson, J.E. (1991) "Histological Effects of Microcystin-LR, a Cyclic Peptide Toxin from the Cyanobacterium (Blue-green Alga) *Microcystis aeruginosa*, on Common Carp (*Cyprinus carpio* L.)", *Aquatic Toxicology*, Vol.20, pp.131-146.

[248] Rangarajan, A., Chellappa, R. and Manjunath, B.S. (1991) "Markov Random Fields and Neural Networks with Applications to Early Vision Problems", in *Artificial Neural Networks and Statistical Pattern Recognition - Old and New Connections*, pp.155-174, I.K. Sethi and A.K. Jain, Eds, Elsevier Science Publishers B.V., Amsterdam.

[249] Ranjithan, S., Eheart, J.W. and Garrett Jr., J.H. (1993) "Neural Network-Based Screening for Groundwater Reclamation Under Uncertainty", *Water Resources Research*, Vol.29, No.3, pp.563-574.

[250] Rao, A.R., Kashyap, R.L. and Mao, L.-T. (1982) "Optimal Choice of Type and Order of River Flow Time Series Models", *Water Resources Research*, Vol.18, No.4, pp.1097-1109.

[251] Rapala, J., Sivonen, K., Luukkainen, R., Niemelä, S.I. (1993) "Anatoxin-a Concentration in *Anabaena* and *Aphanizomenon* Under Different Environmental Conditions and Comparison of Growth by Toxic and Non-Toxic *Anabaena* Strains - a Laboratory Study", *Journal of Applied Phycology*, Vol.5, pp.581-591.

[252] Recknagel, F., French, M., Harkonen, P. and Yabunaka, K.-I. (1995a) "Modelling and Prediction of Algal Blooms: The Artificial Neural Network Approach", *Ecological Modelling*, in press.

[253] Recknagel, F., Hosomi, M., Fukushima, T. and Kong, D.-S. (1995b) "Short-and Long-Term Control of External and Internal Phosphorus Loads in Lakes - A Scenario Analysis", *Water Research*, Vol.29, No.7, pp.1767-1779.

[254] Recknagel, F., Petzoldt, T., Jaeke, O. and Krusche, F. (1994) "Hybrid Expert System DELAQUA - A Toolkit for Water Quality Control of Lakes and Reservoirs", *Ecological Modelling*, Vol.71, pp.17-36.

[255] Repavich, W.M., Sonzogni, W.C., Standridge, J.H., Wedepohl, R.E. and Meisner, L.F. (1990) "Cyanobacteria (Blue-green Algae) in Wisconsin Waters: Acute and Chronic Toxicity", *Water Res*earch, Vol.24, pp.225-231.

[256] Reynolds, C.S. (1984) "*The Ecology of Freshwater Phytoplankton*", University Press, Cambridge, 384p.

[257] Rogers, L.L. and Dowla, F.U. (1994) "Optimization of Groundwater Remediation Using Artificial Neural Networks With Parallel Solute Transport Modeling", *Water Resources Research*, Vol.30, No.2, pp.457-481.

[258] Rosenblatt, F. (1961) "*Principles of Neurodynamics*", Spartan Books, New York.

[259] Rosenblatt, F. (1958) "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain", *Psychological Review*, Vol.65, pp.386-408.

[260] Rositano, J. and Nicholson, B.C. (1994) "Water Treatment Techniques for the Removal of Cyanobacterial Peptide Toxins from Water", *EWS Report*, in press.

[261] Rumelhart, D.E., Hinton G.E. and Williams, R.J. (1986) "Learning Internal Representations by Error Propagation", in *Parallel Distributed Processing*, Vol.1, Ch.8, D.E. Rumelhart and J.L. McClelland, Eds, MIT Press.

[262] Rumelhart, D.E. and Zipser, D. (1985) "Feature Discovery by Competitive Learning", *Cognitive Science*, Vol.9, pp.75-115.

[263] Saha, A. and Keeler, J.D. (1990) "Algorithms for Better Representation and Faster Learning in Radial Basis Function Networks", in *Advances in Neural Information Processing Systems 2*, pp.482-489, D. Touretzky, Ed, Morgan Kaufmann, San Mateo.

[264] Saito, K. and Nakano, R. (1990) "Rule Extraction from Facts and Neural Networks", *Proceedings of the International Neural Network Conference*, Vol.1, Paris, July 9-13, pp.379-382.

[265] Salas, J.D., Boes, D.C. and Smith, R.A. (1982) "Estimation of ARMA Models with Seasonal Parameters", *Water Resources Research*, Vol.18, No.4, pp.1006-1010.

[266] Salas, J.D., Delleur, J.W., Yevjevich, V and Lane, W.L. (1980) "*Applied Modelling of Hydrologic Time Series*", Water Resources Publications, Littleton, Colorado.

[267] Salas, J.D., Tabioas III, G.Q. and Bartolini, P. (1985) "Approaches to Multivariate Modeling of Water Resources Time Series", *Water Resources Bulletin*, Vol.21, No.4, pp.683-708.

[268] Sallas, W.M. and Harville, D.A. (1981) "Best Linear Recursive Estimation for Mixed Linear Models", *Journal of the American Statistical Association*, Vol.76, pp.860-869.

[269] Samad, T. (1988) "Back-Propagation is Significantly Faster if the Expected Value of the Source Unit is Used for Update", *International Neural Network Society Conference Abstracts*.

[270] Schildt, H. (1987) "*Artificial Intelligence Using C*", McGraw Hill, California, pp.12, 71-73.

[271] Schizas, C.N., Pattichis, C.S. and Michaelides, S.C. (1994) "Forecasting Minimum Temperature with Short Time-Length Data Using Artificial Neural Networks", *Neural Network World*, Vol.4, No.2, pp.219-230.

[272] Schmidhuber, J. (1989) "Accelerated Learning in Back-Propagation Nets", in *Connectionism in Perspective*, pp.439-444, R. Pfeiffer, Z. Schreter and F. Fogelman-Soulie, Eds, North-Holland.

[273] Schwarz, G. (1978) "Estimating the Dimension of a Model", *Annals of Statistics*, Vol.6, No.2, pp.461-464.

[274] Schweppe F.C. (1973) "*Uncertain Dynamic Systems*", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 563 p.

[275] Sietsma, J. and Dow, R.J.F. (1991) "Creating Artificial Neural Networks that Generalize", *Neural Networks*, Vol.4, pp.67-79.

[276] Silva, F.M. and Almeida, L.B. (1990) "Acceleration Techniques for the Backpropagation", *Proceedings of the EURASIP Workshop*, Sesimbra, Portugal, February 15-17, pp.110-119.

[277] Sivonen, K. (1990) "Effects of Light, Temperature, Nitrate, Orthophosphate, and Bacteria on Growth of and Hepatotoxin Production by *Oscillatoria agardhii* Strains", *Applied Environmental Microbiology*, Vol.56, pp.2658-2666.

[278] Sivonen, K., Niemalä, S.I., Niemi, R.M., Lepistö, L., Luoma, T.H. and Räsänen (1990) "Toxic Cyanobacteria (Blue-green Algae) in Finnish Fresh and Coastal Waters", *Hydrobiologia*, Vol.190, pp.267-275.

[279] Sivonen, K., Namikoshi, M., Evans, W.R., Carmichael, W.W., Sun, F., Rouhiainen, L., Luukkainen, R. and Rinehart, K.L. (1992a) "Isolation and Characterisation of a Variety of Microcystins from Seven Strains of the Cyanobacterial Genus *Anabaena*", *Applied Environmental Microbiology*, Vol.58, pp.2495-2500.

[280] Sivonen, K., Namikoshi, M., Evans, W.R., Färdig, M., Carmichael, W.W. and Rinehart, K.L. (1992b) "Three New Microcystins, Cyclic Heptapeptide Hepatotoxins, from *Nostoc* sp. Strain 152", *Chem. Res. Toxicol.*, Vol.5, pp.464-469.

[281] Sivonen, K., Namikoshi, M., Evans, W.R., Gromov, B.V., Carmichael, W.W. and Rinehart, K.L. (1992c) "Isolation and Structures of Five Microcystins from

a Russian *Microcystis aeruginosa* Strain CALU 972", *Toxicon*, Vol.30, pp.1481-1485.

[282] Sivonen, K., Skulberg, O.M., Namikoshi, M., Evans, W.R., Carmichael, W.W. and Rinehart, K.L. (1992d) "Two Methyl Ester Derivatives of Microcystins, Cyclic Heptapeptide Hepatotoxins, Isolated from *Anabaena flos-aquae* Strain CYA 83/1", *Toxicon*, Vol.30, pp.1465-1471.

[283] Slatkin, D.N., Stoner, R.D. and Adams, W.H. (1983) "Atypical Pulmonary Thrombosis Caused by a Toxic Cyanobacterial Peptide", *Science*, Vol.220, pp.1383-1385.

[284] Smith, V.H. (1985) "Predictive Models for the Biomass of Blue-Green Algae in Lakes", *Water Resources Bulletin*, Vol.21, No.3, pp.433-439.

[285] Smith, V.H., Willen, B. and Karlsson, B. (1987) "Predicting the Summer Peak Biomass of Four Species of Blue-Green Algae (Cyanophyta/Cyanobacteria) in Swedish Lakes", *Water Resources Bulletin*, Vol.23, No.3, pp.397-402.

[286] Solis, F.J. and Wets, R.J.-B. (1981) "Minimization by Random Search Techniques", *Mathematics of Operations Research*, Vol.6, No.1, pp.19-30.

[287] Specht, D.O. and Shapiro, P.D. (1990) "Training Speed Comparison of Probabilistic Neural Networks with Back-Propagation Networks", *Proceedings of the International Neural Network Conference*, Vol.1, Paris, July 9-13, pp.440-443.

[288] State Water Laboratory (1993) "Toxic Algae", *Biology Unit Fact Sheet BL 11*, July.

[289] Steffensen, D. (1991) "Freshwater Algal Blooms", *Occasional Paper*, Water Resources Management Committee of the Australian Water Resources Council.

[290] Sykora, J.L. and Keleti, G. (1981) "Cyanobacteria and Endotoxins in Drinking Water Supplies", in: *The Water Environment: Algal Toxins and Health*, Carmichael, W.W., Ed, Plenum Press, New York, pp.285-302.

[291] Szu, H. and Hartley, R. (1987) "Fast Simulated Annealing", *Physics Letters* 1222 (3,4), pp.157-162.

[292] Tang, Z., deAlmeida, C. and Fishwick, P.A. (1991) "Time Series Forecasting Using Neural Networks vs. Box-Jenkins Methodology", *Simulation*, Vo.57, No.5, pp.303-310.

[293] Tao, P.C. and Delleur, J.W. (1976) "Seasonal and Nonseasonal ARMA Models", *Journal of the Hydraulics Division*, 102(HY 10), American Society of Civil Engineers, pp.1541-1559.

[294] Taylor, J.G. (1993) "*The Promise of Neural Networks*", Springer Verlag, London.

[295] Thiaville, L., Guillemaud, R. and Niez, J.-J. (1990) "Neural Networks for Edge Detection: A Performance Study" *Proceedings of the International Neural Networks Conference*, Vol.1, Paris, France, pp.63-66.

[296] Thomas, G.A. and Jakeman, A.J. (1983) "Application of Salinity Transport and Accession Models to Water Resources Management", *CRES Working Paper 1983/26.*

[297] Thompstone, R.M., Hipel, K.W. and McLeod, A.I. (1985) "Forecasting Quarter-monthly Riverflow", *Water Resources Bulletin*, Vol.21, No.5, pp.731-741.

[298] Thorpe, S.J. and Imbert, M. (1989) "Biological Constraints on Connectionist Modelling", in *Connectionism in Perspective*, pp.63-93, R. Pfeiffer, Z. Schreter and F. Fogelman-Soulie, Eds, North-Holland.

[299] Tiao, G.C. and Box, G.E.P. (1981) "Modeling Multiple Time Series With Applications", *Journal of the American Statistical Association*, Vol.76, No.376, pp.802-816.

[300] Tiao, G.C. and Tsay, R.S. (1981) "Identification of Nonstationary and Stationary ARMA Models", *Proc. Bus. Econ. Statist. Sec., Amer. Statist. Assoc.*, pp.308-312.

[301] Tirakis, A., Sukissian, L. and Kollias, S. (1990) "An Adaptive Technique for Segmentation and Classification of Textured Images" *Proceedings of the International Neural Networks Conference*, Vol.1, Paris, France, pp.31-34.

[302] Tong, H., Thanoon, B. and Gudmundsson, G. (1985) "Threshold Time Series Modeling of Two Icelandic Riverflow Systems", *Water Resources Bulletin*, Vol.21, No.4, pp.721-730.

[303] Turner, P.C., Gammie, A.J., Hollinrake, K. and Codd, G.A. (1990) "Pneumonia Associated with Cyanobacteria", *British Medical Journal*, Vol.300, pp.1440-1441.

[304] Vaithyanathan, S. and Ignizio, J.P. (1992) "A Stochastic Neural Network for Resource Constrained Scheduling", *Computer and Operations Research*, Vol.19, No.3/4, April/May, pp.241-254.

[305] Vandaele, W. (1983) "*Applied Time Series and Box-Jenkins Models*", Academic Press, Inc., New York, New York.

[306] Varfis, A. and Versino, C. (1990) "Univariate Economic Time Series Forecasting by Connectionist Methods", *Proceedings of the International Neural Network Conference*, Vol.1, Paris, July 9-13, pp.342-345.

[307] Vecchia, A.V. (1985) "Periodic Autoregressive - Moving Average (PARMA) Modeling with Applications to Water Resources", *Water Resources Bulletin*, Vol.21, No.5, pp.683-708.

[308] Vemuri, V. (1988) "Artificial Neural Networks: An Introduction", in *Artificial Neural Networks: Theoretical Concepts*, pp.1-12, V. Vemuri, Ed, Computer Society Press.

[309] Verbyla, A. (1995) Department of Statistics, The University of Adelaide, Personal Communication.

[310] Vogel, R.M. (1986) "The Probability Plot Correlation Coefficient Test for Normal, Lognormal and Gumbel Distribution Hypotheses", *Water Resources Research*, Vol.22, No.4, pp.587-590.

[311] Vollenweider, R.A. (1976) "Advances in Defining Critical Loading Levels for Phosphorus in Lake Eutrophication", *Mem. Ist. Ital. Idrobiol.*, Vol.33, pp.53-83.

[312] Walter, J., Ritter, H. and Schulten, K. (1990) "Non-Linear Prediction With Self-Organizing Maps", *Proceedings of the IJCNN International Joint Conference on Neural Networks*, San Diego, California, June 17-21, pp.I589-I594.

[313] Wang, J. and Changkong, V. (1992) "Recurrent Neural Networks for Linear Programming: Analysis and Design Principles", *Computer and Operations Research*, Vol.19, No.3/4, April/May, pp.297-310.

[314] Wassermann, P.D. (1989) "*Neural Computing: Theory and Practice*", Van Norstrand Reinhold, New York, 230p.

[315] Water Studies Pty. Ltd. (1992) "River Murray Flow Model", Volume 1, Summary Report, Prepared for EWS S.A., EWS Library Reference 92\12.

[316] Wei, W.W.S. (1990) "*Time Series Analysis: Univariate and Multivariate Methods*", Addison-Wesley, Redwood City, California.

[317] Weigend, A.S., Rumelhart, D.E. and Huberman, B.A. (1990) "Predicting the Future: A Connectionist Approach", *International Journal of Neural Systems*, Vol.1,No.3, pp.193-209.

[318] White, R.H. (1990) "The Learning Rate in Back-Propagatin Systems: An Application of Newton's Method", *Proceedings of the IJCNN International Joint Conference on Neural Networks*, San Diego, California, June 17-21, pp.I679-I684.

[319] Wnorowski, A.U. (1992) "Tastes and Odours in the Aquatic Environment: a Review", *Water SA*, Vol.18, pp.203-214.

[320] Wood, G. (1975) "An Assessment of Eutrophication in Australian Inland Waters", *AWRC Technical Paper No.15*, AGPS.

[321] World Health Organisation (1984) "*Guidelines for Drinking Water Quality*", Geneva.

[322] Widrow, B. and Smith, F.W. (1960) "Adaptive Switching Circuits", *IRE WESCON Convention Record*, New York, August 23-26, pp.96-104.

[323] Widrow, B. and Smith, F.W. (1963) "Pattern Recognizing Control Systems", *Computer and Information Sciences Symposium Proceedings*, Spartan Books, Washinngton D.C.

[324] Windsor, C.G. and Harker, A.H. (1990) "Multi-Variate Financial Index Prediction - A Neural Network Study", *Proceedings of the International Neural Network Conference*, Vol.1, Paris, July 9-13, pp.357-360.

[325] Wu, X., Garrett, J.R. and Ghaboussi, J. (1990) "Representation of Material Behaviour: Neural Network-Based Models", *Proceedings of the IJCNN International Joint Conference on Neural Networks*, San Diego, California, June 17-21, pp.I229-I234.

[326] Yeung, D.S. and Fong, H.S. (1994) "Handwritten Chinese Character Recognition by Rule-Embedded Neocognitron", *Neural Computing and Applications*, Vol.2, No.4, pp.216-226.

[327] Young, P.C. (1988) "Recursive Extrapolation, Interpolation and Smoothing of Nonstationary Time Series", *Proceedings of the IFAC Symposium on Identification and System Parameter Estimation*, Beijing, China, pp.33-44.

[328] Young, P.C. (1994) "Time-variable Parameter and Trend Estimation in Non-stationary Economic Time Series", *Journal of Forecasting*, Vol.13, pp.179-210.

[329] Young, P.C., Ng, C.N. and Armitage, P. (1989) "A Systems Approach to Econometric Forecasting and Seasonal Adjustment", *International Journal on Computers and Mathematics with Applications*, Vol.18, pp.481-501.

[330] Young, P.C., Ng, C.N., Lane, K. and Parker, D. (1991) "Recursive Forecasting, Smoothing and Seasonal Adjustment of Non-stationary Environmental Data", *Journal of Forecasting*, Vol.10, pp.57-89.

[331] Yu, S.-Z. (1989) "Drinking Water and Primary Liver Cancer", in: *Primary Liver Cancer*, Tang, Z., Wu, M.C. and Xia, S.S., Eds, China Academic Publishers, Springer Verlag, Berlin, Heidelberg, New York, p.30.

[332] Zhang, S.P., Watanabe, H. and Yamada, R. (1994) "Prediction of Daily Water Demands by Neural Networks",in *Stochastic and Statistical Methods in Hydrology and Environmental Engineering, Vol.3, Time Series Analysis in*

Hydrology and Environmental Engineering, pp.217-227, K.W. Hipel, A.I. McLeod, U.S. Panu and V.P. Singh, Eds, Kluwer Academic Publishers, Dordrecht, Netherlands.

[333] Zhu, M.-L. and Fujita, M. (1994) "Application of Neural Networks to Runoff Prediction", in *Stochastic and Statistical Methods in Hydrology and Environmental Engineering, Vol.3, Time Series Analysis in Hydrology and Environmental Engineering*, pp.205-216, K.W. Hipel, A.I. McLeod, U.S. Panu and V.P. Singh, Eds, Kluwer Academic Publishers, Dordrecht, Netherlands.

[334] Zillberg, B. (1966) "Gastroenteritis in Salisbury European Children - a Five Year Study", *Central African Journal of Medicine*, Vol.12, No.9, pp.164-168.

Corrigenda for

Use of Artificial Neural Networks for Modelling Multivariate Water Quality Time Series

by

Holger R. Maier

On page 7, the sentence:

The advantages of using artificial neural networks include (Jones and Hoskins, 1987; Lippmann, 1987; Daniell, 1991; Burke and Ignizio, 1992; Burke, 1991; NeuralWare, Inc., 1991; Kimoto and Asakawa, 1990; Josin, 1987; Saito and Nakano, 1990; Hecht-Nielsen, 1988; Maren et al., 1990; Hubick, 1992):

should be replaced with:

The advantages of using artificial neural networks include (Jones and Hoskins, 1987; Lippmann, 1987; Daniell, 1991; Burke and Ignizio, 1992; Ignizio Burke, 1991; NeuralWare, Inc., 1991; Kimoto and Asakawa, 1990; Josin, 1987; Saito and Nakano, 1990; Hecht-Nielsen, 1988; Maren et al., 1990; Hubick, 1992):

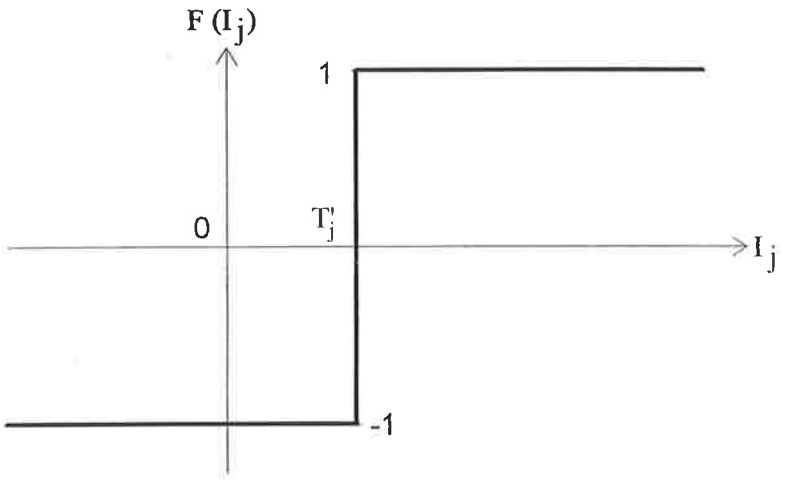
On page 13, the notation:

x_i = the input from node i, i = 0, 1, ..., N^P

should be replaced with:

x_i = the input from node i, i = 1, 2, ..., N^P

Figure 2.4 on page 15 should be replaced with the following figure:



On page 26, the sentence:

Adaptive resonance theory (ART) networks (described in Lippmann, 1987; Wassermann, 1989; Maren et al., 1990), bi-directional associative memory (BAM) networks (Kosko, 1987) and recirculation networks (Hinton, 1988) are examples of this type of network.

should be replaced with:

Adaptive resonance theory (ART) networks (described in Lippmann, 1987; Wassermann, 1989; Maren et al., 1990), bi-directional associative memory (BAM) networks (Kosko, 1987) and recirculation networks (Hinton and McClelland, 1988) are examples of this type of network.

On page 39, the sentence:

Weigend et al. (1990, 1991) suggest using the following heuristic rule for determining the initial number of hidden nodes:

should be replaced with:

Weigend et al. (1990) suggest using the following heuristic rule for determining the initial number of hidden nodes:

On page 85, the sentence:

If d^{DW} exceeds a critical value, the model is deemed to be inadequate.

should be replaced with:

If d^{DW} exceeds a critical value, the residuals are autocorrelated and hence one of the basic assumptions of the model is violated.

On page 138, the list:

Mannum, 1989, days 64 and 100.
Morgan, 1989, day 116.
Morgan, 1990, day 305.

should be replaced with:

Mannum, 1988, day 120.
Mannum, 1989, days 64 and 100.
Morgan, 1987, days 64 and 170.
Morgan, 1989, day 116.
Morgan, 1990, day 305.

On page 155, the sentences:

There seems to be no appreciable lag between the two flow records. This is partly a function of the fact that lock operators change the lock configuration so that a constant weir pool level is maintained. As a result, changes in flow propagate downstream very quickly.

should follow the sentence:

There also seems to be quite a strong inverse relationship between flow and salinity at Murray Bridge.

On page 156, the sentence:

The means, standard deviations, maxima and minima of the river level time series are shown in Table 3.5.

should be replaced with:

The means, standard deviations, maxima and minima of the river level time series (which are given above the Australian Height Datum (AHD)) are shown in Table 3.5.

On page 526, reference [4] should be replaced with:

Adeli, H. and Yeh, L. (1990) "Neural Network Learning in Engineering Design", *Proceedings of the International Neural Network Conference*, Vol.1, Paris, July 9-13, pp.412-415.

On page 527, reference [8] should follow reference [14].

On page 527, reference [12] should be replaced with:

Ansley, C.F. and Kohn, R. (1984) "On the Estimation of ARIMA Models with Missing Values", in *Time Series Analysis of Irregularly Observed Data*, E. Parzen, Ed, Springer-Verlag, New York.

On page 528, reference [19] should be replaced with:

Bergman, M.J. and Delleur, J.W. (1985a) "Kalman Filter Estimation and Prediction of Daily Stream Flows: I. Review, Algorithm, and Simulation Experiments", *Water Resources Bulletin*, Vol.21, No.5, pp.827-832.

On page 528, reference [27] should be replaced with:

Box, G.E.P. and Pierce, D.A. (1970) "Distribution of Residual Auto-correlations in Autoregressive-Integrated Moving Average Time-series Models", *J. Amer. Statist. Ass.*, Vol.65, pp.1509-1526.

On page 529, reference [33] should be replaced with:

Buringh, P. (1979) "Food Production Potential of the World", in: *The World Food Problem; Concensus and Conflict*, Radhe Sinha, Ed, Pergamom Press, New York, pp.477-485.

On page 533, reference [76] should follow reference [77].

Reference [81] on page 534 should follow reference [77] but before reference [76] on page 533.

On page 534, reference [79] should follow reference should follow reference [80].

On page 534, reference [87] should be replaced with:

Galey, F.D., Beasley, V.R., Carmichael, W.W., Kleppe, G., Hooser, S.B. and Haschek, W.M. (1987) "Blue-green Algae (*Microcystis aeruginosa*) Hepatotoxicosis in Dairy Cows", *American Journal of Veterinary Research*, Vol.48, pp.1415-1420.

On page 537, reference [121] should be replaced with:

Hecht-Nielsen, R. (1988a) "Applications of Counterpropagation Networks", *Neural Networks*, Vol.1, pp.131-139.

On page 538, reference [122] should be replaced with:

Hecht-Nielsen, R. (1988b) "Neurocomputing: Picking the Human Brain", *IEEE Spectrum*, Vol. 25(3), March, pp.36-41.

On page 539, reference [135] should follow reference [136].

On page 542, reference [170] should follow reference [171].

On page 543, reference [179] should be replaced with:

Kohonen, T. (1988) "Statistical Pattern Recognition with Neural Networks: Benchmark Studies", *Proceedings of the Second Annual IEEE International Conference on Neural Networks*, Vol.1.

On page 546, reference [202] should follow reference [203].

On page 547, reference [217] should follow reference [218].

On page 548, reference [223] should be replaced with:

Murray-Darling Basin Commission (MDBC) (1993) "Technical Advisory Group Outcomes" MDBC, Canberra.

On page 548, reference [225] should be replaced with:

Murray-Darling Basin Ministerial Council (MDBMC) (1993) "Algal Management Strategy for the Murray-Darling Basin (Draft)", MDBC, Canberra.

On page 548, reference [229] should follow reference [226].

On page 549, reference [234] should follow reference [235].

On page 550, reference [244] should follow reference [245].

On page 553, reference [278] should follow reference [281].

On page 557, the following reference should be added after reference [313]:

Wann, M., Hediger, T. and Greenbaum, N.N. (1990) "The Influence of Training Sets on Generalisation in Feedforward Neural Networks", *Proceedings of the IEEE/International Joint Conference on Neural Networks*, Vol.III, pp.137-142.

On page 557, reference [322] should be replaced with:

Widrow, B. and Hoff, M. (1960) "Adaptive Switching Circuits", *IRE WESCON Convention Record*, New York, August 23-26, pp.96-104.

References [322] to [324] on pages 557 and 558 should follow reference [318] on page 557.
