Contents

Certificate ii

Declaration by the Candidate iii

Copyright Transfer Certificate iv

Acknowledgements vii

Contents ix

List of Figures xv

List of Tables xix

Abbreviations xxi

Symbols xxiii

Preface xxv

1 Introduction 1

1.1 Introduction 1

1.2 Time Series Analysis 3

1.2.1 Univariate Time Series 3

1.2.2 Modelling Time Series 4

1.2.2.1 Autoregressive Model 4

1.2.2.2 Moving Average Model 5

1.2.2.3 ARMA Model 5

1.2.3 Stationary Time Series 6

1.2.3.1 Augmented Dickey-Fuller Test 6
1.2.3.2 Autocorrelation Function Plots. 7
1.2.4 ARIMA Model[197] 7
1.2.4.1 Calculating Model Parameter for ARIMA 8
1.3 Motivation 9
1.4 Clones, Bugs and Failures 11
1.5 Thesis Objectives 12
1.6 Thesis Contribution 13
1.6.1 Temporal Bug Pattern Prediction 13
1.6.2 Clone Evolution Prediction 13
1.6.3 Software Reliability Prediction Using TBF 14
1.7 Thesis Organization 14

2 Literature Review **17**
2.1 Introduction 17
2.2 Literature Review (Software Bug Prediction) 17
2.3 Literature Review (Clone Detection and Clone Evolution Prediction) 51
2.4 Literature Review (Software Reliability Prediction) 59
2.5 Conclusion 71

3 Temporal Bug Pattern Prediction **73**
3.1 Introduction 73
3.2 Dataset Description 74
3.2.1 Debian 75
3.2.2 Mozilla Firefox 75
3.2.3 Eclipse 76
3.3 Problem Description 76
3.4 Neural Network Approach for Bug Number Prediction 78
3.4.1 Neural Network Model 79
3.4.1.1 Transfer Function 79
3.4.1.2 Heuristic Information about Bug Patterns 80
3.4.2 Data Preparation 82
3.4.3 Proposed System 83
3.4.4 Implementation 83
3.4.5 Evaluation 85
3.4.6 Results and Discussion 85
3.4.6.1 Effect of Transfer Function 86
3.4.6.2 Effect of Number of Hidden Layer Neurons 89
3.4.6.3 Effect of Number of Input Layer Neurons 90
3.4.6.4 Effect of Heuristic Information 90
3.5 A Hybrid Technique for Debian Bug Number Prediction 92

3.5.1 The Hybrid Model . . . 92
3.5.2 Data Preparation . . . 95
3.5.3 Evaluation . . . 95
3.5.4 Implementation and Result . . . 95
3.5.4.1 ARIMA . . . 96
3.5.4.2 Artificial Neural Network (ANN) . . . 97
3.5.4.3 Hybrid Model (ARIMA + ANN) . . . 98
3.6 An Ensemble Technique for Bug Number Prediction . . . 99
3.6.1 The Ensemble Model . . . 100
3.6.2 Data Preparation . . . 102
3.6.3 Evaluation . . . 102
3.6.4 Implementation and Result . . . 102
3.6.4.1 ARIMA . . . 102
3.6.4.2 ANN Model . . . 103
3.6.4.3 Ensemble Model . . . 103
3.7 Prediction of Temporal Bug Patterns of Debian Using Markov Model . . 105
3.7.1 Markov Chain Model . . . 106
3.7.1.1 Conversion from Higher Order to 1st Order . . . 106
3.7.2 Data Preparation . . . 107
3.7.2.1 Stationarity Test . . . 109
3.7.3 Generation of Markov Model . . . 110
3.7.3.1 Formation of Markov States . . . 110
3.7.3.2 K- Median clustering . . . 113
3.7.4 Distribution of Markov States . . . 114
3.7.5 Formation of Transition Matrices . . . 115
3.7.6 Evaluation . . . 119
3.7.7 Results and Interpretation . . . 120
3.7.7.1 Evaluation based on number of Training Instances . . . 120
3.7.7.2 Evaluation based on Higher Order Markov Chain . . . 121
3.7.7.3 Evaluation Based upon Size of Prediction Window . . . 122
3.8 Analysis of Temporal Bug Patterns in Open Source Software Using Hidden Markov Model[169] . . . 123
3.8.1 Hidden Markov Model . . . 124
3.8.1.1 The Basic Problems of HMM . . . 124
3.8.2 Data Preparation . . . 125
3.8.3 Model Building . . . 125
3.8.4 Model Implementation . . . 128
3.8.4.1 HMM Model 1 . . . 128
3.8.4.2 HMM Model 2 . . . 129
3.8.4.3 HMM Model 3 . . . 130
3.8.5 Results and Interpretation . . . 131

3.8.5.1 Most probable state sequence 131
3.8.5.2 Most frequent observation sequence 134
3.9 Conclusion 135

Clone Evolution Prediction **137**
3.10 Introduction 137
3.11 Software Clone Types 138
3.12 Dataset Description 139
3.13 Clone Detection Process 139
3.13.1 Pre-Processing 140
3.13.2 Transformation 140
3.13.3 Clone Extraction 141
3.13.4 Normalization 141
3.13.5 Match detection 141
3.13.6 Clone Detection Tools Used 142
3.13.6.1 CloneDr 142
3.14 Problem Description 143
3.15 Hybrid Modeling Approach for Software Clone Evolution Prediction . . . 144
3.15.1 The Hybrid Model 145
3.15.2 Evaluation 146
3.15.3 Data Preparation 147
3.15.4 Implementation and Result 147
3.15.4.1 ARIMA 148
3.15.4.2 Hybrid Model 148
3.16 Temporal Analysis of Software Clone Evolution using Software Metrics[163]151
3.16.1 Software Metrics and Software Clones 151
3.16.1.1 Software Metrics Extraction 152
3.16.2 Design of Experiments 152
3.16.2.1 Linear regression 153
3.16.2.2 Multi-Layer Perceptron 153
3.16.3 Implementation and Result 154
3.16.3.1 First Model 155
3.16.3.2 Second Model 156
3.16.3.3 Third Model 156
3.16.4 Evaluation and Interpretation 157
3.16.4.1 Evaluation of First Model 158
3.16.4.2 Evaluation of Second Model 158
3.16.4.3 Evaluation of Third Model 159
3.16.5 Interpretation of the Models 161
3.17 A Comparison among ARIMA, BP-NN, and MOGA-NN for Software Clone Evolution Prediction[162] 163
3.17.1 MOGA-NN for Software Clone Evolution Prediction 163

3.17.1.1 Prediction Intervals . . . 165
3.17.1.2 NSGA-II [50] . . . 166
3.17.2 Design of Experiments . . . 167
3.17.2.1 ARIMA Modelling . . . 167
3.17.2.2 Back Propagation based Learning . . . 169
3.17.2.3 Multi-Objective Genetic Algorithm Based Neural Network Learning . . . 169
3.17.3 Evaluation and Interpretation . . . 173
3.17.4 ARIMA Model Evaluation . . . 174
3.17.5 Back Propagation based NN Evaluation . . . 174
3.17.6 MOGA-NN Evaluation . . . 175
3.18 Conclusion . . . 183

Software Reliability Prediction **185**
3.19 Introduction . . . 185
3.20 Dataset Description . . . 186
3.21 Problem Description . . . 187
3.22 A Hybrid Technique for Software Reliability Prediction . . . 188
3.22.1 The Hybrid Model . . . 188
3.22.2 Evaluation . . . 189
3.22.3 Implementation and Result . . . 191
3.22.3.1 ARIMA . . . 191
3.22.3.2 Hybrid Model . . . 191
3.23 A Prediction Interval Based Estimation Approach for Software Reliability Modelling . . . 194
3.23.1 MOGA-NN Modelling . . . 194
3.23.1.1 Prediction Intervals . . . 196
3.23.1.2 NSGA-II[50] . . . 197
3.23.2 ELM+KNN Modelling . . . 200
3.24 Implementation and Result . . . 204
3.24.0.1 MOGA-NN Modelling . . . 204
3.24.0.2 ELM+KNN Modelling . . . 207
3.24.1 Evaluation and Interpretation . . . 209
3.24.1.1 MOGA - NN Evaluation . . . 210
3.24.1.2 ELM + KNN Evaluation . . . 210
3.24.2 External Validity Testing . . . 218
3.25 Conclusion . . . 220

Conclusion **221**
3.26 Future Scope of This Thesis . . . 223

List of Publications **225**
.1 Journal Publication 225
.2 Conference Publication 226
.3 Paper Under Review 226

DataSet Used In This Thesis **227**
.4 Dataset Description 227

Bibliography **229**