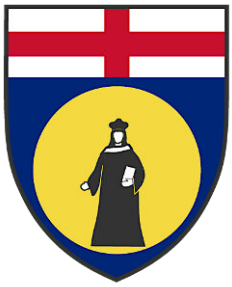
UNIVERSITY OF GENOA

DEPARTMENT OF ECONOMICS

PhD Program in Economics and Quantitative Methods
XXXVIII Cycle



Forecasting in Industry and Financial Context

Candidate: Giacomo Gaggero

Supervisors: Pier Giuseppe Giribone, Claudio Piga



Borsa di dottorato cofinanziata con risorse dell'Unione europea-NextGeneration EU
Piano Nazionale di Ripresa e Resilienza la Missione 4, componente 2
("Dalla Ricerca all'Impresa")

Certificate of Authorship and Co-authorship

This section contains an overview of the role of the co-authors in the realization of this thesis.

1. Demand Forecasting, Manufacturing, Data Analysis

This study was carried out on behalf of, and in collaboration with, VHit (https://vhit-weifu.com/), the sponsor of my PhD programme. For VHit, I was appointed with implementationg various forecasting methods to predict the demand for their products. In conducting this study, I was assisted by Professor Pier Giuseppe Giribone (University of Genoa), who contributed to the implementation of the SARIMA and LSTM models. My focus was on data cleaning, the classification process, and the forecasting of lumpy series using TSB methods.

2. Improving the Performance of Traditional Interest Rates Term Structure Models

This study was initially developed by myself and my supervisor, Professor Pier Giuseppe Giribone (University of Genoa). Following an initial study focused on developed countries' yield curves, a spin-off was created focusing on the yield curves of Asia-Pacific countries, where we were joined by Professor Duccio Martelli (University of Perugia). Subsequently, during my visiting period at Keele University, a second spin-off was carried out, focusing on the yield curves of the main sovereign countries in the Euro area, collaborating with Professor Gabriella Legrenzi (University of Keele).

In this study, my main contribution was the parameter calibration and performance analysis, under the supervision of Professor Giribone, who aided me in the theoretical understanding of the models and algorithms used. Professor Duccio Martelli and Professor Legrenzi provided insights regarding the possible practical applications and the problems that this study aims to solve.

3. A reliable and prudent forward-looking Value at Risk estimation

This study was conducted in collaboration with Professor Pier Giuseppe Giribone (University of Genoa) and Sanmoy Mukerjee (University of Genoa). My co-authors focused on the models necessary to determine the drift (ARIMA, LSTM). My work concentrated on volatility estimation, option pricing, portfolio forecasting using the Monte Carlo method, and VaR estimation.

Preface

This thesis is the result of my PhD journey. The research was carried out during the course of Economics and Quantitative Methods at the University of Genoa and during my visiting period at Keele University.

My works focused mainly on Forecasting applied across diverse contexts and using various methodologies, a theme that served as the core of my PhD path.

The first chapter is the result of a collaboration between the University of Genoa and VHit, an Italian manufacturing company that sponsored my PhD programme. We worked together to design a forecasting process for the products sold by the company, aiming to predict future demand. The results highlight that in the manufacturing context, demand is often highly intermittent and necessitates the use of specialized forecasting models for this task. These specialized methods must be integrated with other forecasting techniques to ensure a variety of approaches are available, depending on the specific characteristics of the products' time series.

The second chapter explores whether it is possible to improve the yield curve fitting performance of traditional parametric models through the aid of nature-inspired optimization techniques. The results show that the performance improves to such an extent that it matches the results obtained with Machine Learning techniques, while retaining the significant advantage of preserving the economic interpretability inherent in parametric models.

The third chapter investigates how to create a particularly prudent Value at Risk (VaR) estimation by utilizing multiple techniques to calculate the necessary parameters. This approach is then applied to a portfolio of instruments with a non-linear payoff. To do this, we chose the options. The results demonstrate how the ability to select the most conservative values among the different methods guarantees an additionally prudent and reliable VaR estimation.

Contents

Certificate of Authorship and Co-authorship i

Preface iii

Chapter 1: Demand Forecasting for a Manufacturing Company 1

Abstract 2

1 Introduction 4

2 Literature Review 5
2.1 Classification . . . 5
2.1.1 Time series classification . . . 6
2.2 Intermittent Demand Forecasting . . . 8

3 Data 12

4 Classification 17
4.1 Time Series Classification Methodology . . . 17
4.2 Classification Results . . . 19
4.3 Revenue Impact Analysis by Classification . . . 21

5 Forecasting 23
5.1 Forecasting Intermittent Time Series . . . 24
5.2 Forecasting Stationary Time Series . . . 26
5.3 Forecasting Non-Stationary Time Series . . . 29

6 Results 31
6.1 Performance of the TSB model . . . 31
6.2 Performance of the SARIMA model . . . 33
6.3 Performance of the RNN model . . . 34

7 Conclusions 36

8 Appendix 40
8.1 Code Explanation . . . 40
8.1.1 Data Cleaning Process . . . 40
8.1.2 General Statistics . . . 42
8.1.3 Classification . . . 43
8.1.4 Forecasting Methods . . . 45
8.1.5 Forecasting Validation . . . 47

8.1.6 Comprehensive Results 48

Chapter 2: Improving the Performance of Traditional Interest Rates Term Structure Models Using Nature-Inspired Optimization Algorithms **50**

Abstract **51**

1 Introduction **53**

2 Literature Review **55**

3 Methodology **58**
3.1 Financial Bootstrapping 58
3.2 Nonlinear Parametric Models 60
3.2.1 Nelson-Siegel model (1987) 60
3.2.2 Svensson model (1994) 61
3.2.3 De Rezende-Ferreira-Ferreira model (2011) 62
3.3 Optimizers 62
3.3.1 Levenberg-Marquardt algorithm 63
3.3.2 Trust Region algorithm 64
3.4 Memetic Technique 66
3.5 Nature-Inspired algorithms 66
3.5.1 Genetic Algorithms 67
3.5.2 Particle Swarm Optimization 68
3.5.3 Simulated Annealing 69
3.6 Gaussian Process approach 70
3.6.1 Extension via Basis Functions and Latent Gaussian Processes 71
3.6.2 Parameter Estimation and Prediction 72
3.7 Forecasting using Vasicek dynamics 73
3.7.1 Stochastic Differential Equation (SDE) 73
3.7.2 Exact solution and simulation 74
3.7.3 Parameter Estimation 74
3.7.4 Zero-coupon bond pricing 75

4 Applications **77**
4.1 Main Western Countries 78
4.1.1 Data 78
4.1.2 Results 78
4.2 Main Pacific Economies 81
4.2.1 Data 81
4.2.2 Results 83
4.3 Sovereign Yield Curves in European Countries 87
4.3.1 Data 87
4.3.2 Results 88

5 Conclusions **91**

A Appendix **97**

A.1 Main Western Countries . . . 97
A.1.1 IR term structures (Developed Economies) . . . 97
A.1.2 Base Models (Developed Economies) . . . 100
A.1.3 Optimized Models (Developed Economies) . . . 108
A.1.4 GPR and further Results (Developed Economies) . 116
A.2 Main Pacific Economies . . . 120
A.2.1 IR term structures (Pacific Economies) . . . 120
A.2.2 Base Models (Pacific Economies) . . . 125
A.2.3 Optimized Models (Pacific Economies) . . . 139
A.2.4 Additional Results (Pacific Economies) . . . 180
A.3 Sovereign Yield Curves (European Countries) . . . 184
A.3.1 Performance Comparison (Base vs. Optimized) . . 184
A.3.2 Cumulative Performance Comparison . . . 208
A.3.3 Global Performance . . . 216
A.3.4 Vasicek Forecasts . . . 218

Chapter 3: A reliable and prudent forward-looking Value at Risk estimation applied to an Option Portfolio **221**

Abstract **222**

1 Introduction **224**

2 Literature Review **226**

3 Data **228**
3.1 Stock Characteristics . . . 228
3.2 Option Characteristics . . . 230
3.2.1 Option formulas . . . 230
3.2.2 Deal Caratteristics . . . 237

4 Methodology **239**
4.1 Monte Carlo Simulation Framework and Contribution . . 239
4.2 Multiple Estimation Methodologies . . . 240
4.2.1 Volatility (σ) Estimation Models . . . 240
4.2.2 Drift (μ) and Dividend (q) Estimation Models . . . 242
4.3 Correlation Analysis . . . 243

5 Results **245**
5.1 Parameter Selection . . . 245
5.2 Monte Carlo Simulation . . . 248
5.3 VaR Estimation . . . 248

6 Conclusions **250**

A Appendix: R Code Implementation **254**
A.1 Initialization and Data Loading . . . 254
A.2 Volatility Estimation . . . 254
A.3 Dividend Yield Estimation . . . 255
A.4 Drift Estimation . . . 255

A.5 Correlation Estimation and Cholesky Decomposition . . 256
A.6 Option Definition 256
A.7 Multi-Scenario Monte Carlo Simulation 256
A.8 Prudential Aggregation 257
A.9 VaR Computation 257

List of Figures

Chapter 1: Demand Forecasting for a Manufacturing Company **1**

2.1 Demand time series 8

3.1 Cleaning Process 15
3.2 Pareto analysis of Family Products 16
3.3 Pareto analysis of Family Product Clients 16

4.1 Classification of time series for Family Product . . . 19
4.2 Classification of time series for Family Product Client 20
4.3 Pareto analysis by classification for Family Product . 21
4.4 Pareto analysis by classification for Family Product Client 21
4.5 Count share by clas- sification for Family Product . . 22
4.6 Count share by classification for Family Product Client 22
4.7 Revenue share by classification for Family Product . 22
4.8 Revenue share by classification for Family Product Client 22

5.1 Overview of the forecasting methodology selection process 23
5.2 Example of TSB model forecast adjustment over time 26
5.3 Example of SARIMA model forecast adjustment over time 28
5.4 Example of RNN model forecast adjustment over time 30

6.1 TSB model validation example - Family Product . . 32
6.2 Heatmap of forecasting errors for the TSB model . . 32
6.3 Level of accuracy of predictions for the TSB model . 32
6.4 SARIMA model validation example - Family Product 33
6.5 Heatmap of forecasting errors for the SARIMA model 33
6.6 level of accuracy of predictions for the SARIMA model 34
6.7 RNN model validation example - Family Product . . 34
6.8 Heatmap of forecasting errors for the RNN model . 35
6.9 Level of accuracy of predictions for the RNN model 35

Chapter 2: Improving the Performance of Traditional Interest Rates Term Structure Models Using Nature-Inspired Optimization Algorithms **50**

4.1 Average of fitting performance for different currencies 79
4.2 In-Sample RMSE Time Series for CHF, EUR, GBP, JPY, and USD 80
4.3 Out of-Sample RMSE Time Series for CHF, EUR, GBP, JPY, and USD 81
4.4 The analyzed Pacific area countries 82
4.5 Granulometry and securities used for the zero rates bootstrap 82
4.6 Average of fitting performance for different currencies 86
4.7 the European countries analysed in the study 87
4.8 Comparative performance of parametric models with and without optimization algorithms 88
4.9 Relative computational speed of the different optimization algorithms 89
4.10 Forecast of the 30-year interest rates curve for Italian government bonds on July 1, 2025 (10 simulations) . 90

A.1 Interest rates term structure – CHF 97
A.2 Interest rates term structure – EUR 98
A.3 Interest rates term structure – GBP 98
A.4 Interest rates term structure – JPY 99
A.5 Interest rates term structure – USD 99
A.6 Nelson-Siegel model coefficients – CHF 100
A.7 Nelson-Siegel model coefficients – EUR 100
A.8 Nelson-Siegel model coefficients – GBP 101
A.9 Nelson-Siegel model coefficients – JPY 101
A.10 Nelson-Siegel model coefficients – USD 102
A.11 Svensson model coefficients – CHF 102
A.12 Svensson model coefficients – EUR 103
A.13 Svensson model coefficients – GBP 103
A.14 Svensson model coefficients – JPY 104
A.15 Svensson model coefficients – USD 104
A.16 De Rezende-Ferreira model coefficients – CHF . . . 105
A.17 De Rezende-Ferreira model coefficients – EUR . . . 105
A.18 De Rezende-Ferreira model coefficients – GBP 106
A.19 De Rezende-Ferreira model coefficients – JPY 106
A.20 De Rezende-Ferreira model coefficients – USD . . . 107
A.21 Nelson-Siegel model (GA-optimized) coefficients – CHF 108
A.22 Nelson-Siegel model (GA-optimized) coefficients – EUR 108
A.23 Nelson-Siegel model (GA-optimized) coefficients – GBP 109
A.24 Nelson-Siegel model (GA-optimized) coefficients – JPY 109
A.25 Nelson-Siegel model (GA-optimized) coefficients – USD 110
A.26 Svensson model (GA-optimized) coefficients – CHF 110

A.27 Svensson model (GA-optimized) coefficients – EUR 111
A.28 Svensson model (GA-optimized) coefficients – GBP 111
A.29 Svensson model (GA-optimized) coefficients – JPY . 112
A.30 Svensson model (GA-optimized) coefficients – USD 112
A.31 De Rezende-Ferreira model (PSO-optimized) coefficients – CHF 113
A.32 De Rezende-Ferreira model (PSO-optimized) coefficients – EUR 113
A.33 De Rezende-Ferreira model (PSO-optimized) coefficients – GBP 114
A.34 De Rezende-Ferreira model (PSO-optimized) coefficients – JPY 114
A.35 De Rezende-Ferreira model (PSO-optimized) coefficients – USD 115
A.36 In-sample RMSE time series – GPR model (all currencies) 116
A.37 Out-of-sample RMSE time series – GPR model (all currencies) 116
A.38 Average performance of all methods 117
A.39 Cumulative R^2 bounds – CHF 117
A.40 Cumulative R^2 bounds – EUR 118
A.41 Cumulative R^2 bounds – GBP 118
A.42 Cumulative R^2 bounds – JPY 119
A.43 Cumulative R^2 bounds – USD 119
A.44 Interest rates term structure – AUD 120
A.45 Interest rates term structure – CNY 121
A.46 Interest rates term structure – HKD 121
A.47 Interest rates term structure – INR 122
A.48 Interest rates term structure – JPY 122
A.49 Interest rates term structure – MYR 123
A.50 Interest rates term structure – NZD 123
A.51 Interest rates term structure – TWD 124
A.52 Interest rates term structure – USD 124
A.53 Nelson model coefficients – AUD 125
A.54 Nelson model coefficients – CNY 126
A.55 Nelson model coefficients – HKD 126
A.56 Nelson model coefficients – INR 127
A.57 Nelson model coefficients – JPY 127
A.58 Nelson model coefficients – MYR 128
A.59 Nelson model coefficients – NZD 128
A.60 Nelson model coefficients – TWD 129
A.61 Nelson model coefficients – USD 129
A.62 Svensson model coefficients – AUD 130
A.63 Svensson model coefficients – CNY 130
A.64 Svensson model coefficients – HKD 131
A.65 Svensson model coefficients – INR 131
A.66 Svensson model coefficients – JPY 132
A.67 Svensson model coefficients – MYR 132

A.68 Svensson model coefficients – NZD 133
A.69 Svensson model coefficients – TWD 133
A.70 Svensson model coefficients – USD 134
A.71 De Rezende-Ferreira model coefficients – AUD . . . 134
A.72 De Rezende-Ferreira model coefficients – CNY . . . 135
A.73 De Rezende-Ferreira model coefficients – HKD . . . 135
A.74 De Rezende-Ferreira model coefficients – INR 136
A.75 De Rezende-Ferreira model coefficients – JPY 136
A.76 De Rezende-Ferreira model coefficients – MYR . . . 137
A.77 De Rezende-Ferreira model coefficients – NZD . . . 137
A.78 De Rezende-Ferreira model coefficients – TWD . . . 138
A.79 De Rezende-Ferreira model coefficients – USD . . . 138
A.80 Nelson-Siegel model (GA-optimized) coefficients – AUD 139
A.81 Nelson-Siegel model (GA-optimized) coefficients – CNY 139
A.82 Nelson-Siegel model (GA-optimized) coefficients – HKD 140
A.83 Nelson-Siegel model (GA-optimized) coefficients – INR 140
A.84 Nelson-Siegel model (GA-optimized) coefficients – JPY 141
A.85 Nelson-Siegel model (GA-optimized) coefficients – MYR 141
A.86 Nelson-Siegel model (GA-optimized) coefficients – NZD 142
A.87 Nelson-Siegel model (GA-optimized) coefficients – TWD 142
A.88 Nelson-Siegel model (GA-optimized) coefficients – USD 143
A.89 Nelson-Siegel model (PSO-optimized) coefficients – AUD 143
A.90 Nelson-Siegel model (PSO-optimized) coefficients – CNY 144
A.91 Nelson-Siegel model (PSO-optimized) coefficients – HKD 144
A.92 Nelson-Siegel model (PSO-optimized) coefficients – INR 145
A.93 Nelson-Siegel model (PSO-optimized) coefficients – JPY 145
A.94 Nelson-Siegel model (PSO-optimized) coefficients – MYR 146
A.95 Nelson-Siegel model (PSO-optimized) coefficients – NZD 146
A.96 Nelson-Siegel model (PSO-optimized) coefficients – TWD 147
A.97 Nelson-Siegel model (PSO-optimized) coefficients – USD 147

A.98 Nelson-Siegel model (SA-optimized) coefficients – AUD . . . 148
A.99 Nelson-Siegel model (SA-optimized) coefficients – CNY . . . 148
A.100 Nelson-Siegel model (SA-optimized) coefficients – HKD . . . 149
A.101 Nelson-Siegel model (SA-optimized) coefficients – INR . . . 149
A.102 Nelson-Siegel model (SA-optimized) coefficients – JPY . . . 150
A.103 Nelson-Siegel model (SA-optimized) coefficients – MYR . . . 150
A.104 Nelson-Siegel model (SA-optimized) coefficients – NZD . . . 151
A.105 Nelson-Siegel model (SA-optimized) coefficients – TWD . . . 151
A.106 Nelson-Siegel model (SA-optimized) coefficients – USD . . . 152
A.107 Svensson model (GA-optimized) coefficients – AUD 152
A.108 Svensson model (GA-optimized) coefficients – CNY 153
A.109 Svensson model (GA-optimized) coefficients – HKD 153
A.110 Svensson model (GA-optimized) coefficients – INR . 154
A.111 Svensson model (GA-optimized) coefficients – JPY . 154
A.112 Svensson model (GA-optimized) coefficients – MYR 155
A.113 Svensson model (GA-optimized) coefficients – NZD 155
A.114 Svensson model (GA-optimized) coefficients – TWD 156
A.115 Svensson model (GA-optimized) coefficients – USD 156
A.116 Svensson model (PSO-optimized) coefficients – AUD 157
A.117 Svensson model (PSO-optimized) coefficients – CNY 157
A.118 Svensson model (PSO-optimized) coefficients – HKD 158
A.119 Svensson model (PSO-optimized) coefficients – INR 158
A.120 Svensson model (PSO-optimized) coefficients – JPY 159
A.121 Svensson model (PSO-optimized) coefficients – MYR 159
A.122 Svensson model (PSO-optimized) coefficients – NZD 160
A.123 Svensson model (PSO-optimized) coefficients – TWD 160
A.124 Svensson model (PSO-optimized) coefficients – USD 161
A.125 Svensson model (SA-optimized) coefficients – AUD 161
A.126 Svensson model (SA-optimized) coefficients – CNY 162
A.127 Svensson model (SA-optimized) coefficients – HKD 162
A.128 Svensson model (SA-optimized) coefficients – INR . 163
A.129 Svensson model (SA-optimized) coefficients – JPY . 163
A.130 Svensson model (SA-optimized) coefficients – MYR 164
A.131 Svensson model (SA-optimized) coefficients – NZD 164
A.132 Svensson model (SA-optimized) coefficients – TWD 165
A.133 Svensson model (SA-optimized) coefficients – USD . 165
A.134 De Rezende-Ferreira model (GA-optimized) coefficients – AUD . . . 166

A.135 De Rezende-Ferreira model (GA-optimized) coefficients – CNY . . . 166
A.136 De Rezende-Ferreira model (GA-optimized) coefficients – HKD . . . 167
A.137 De Rezende-Ferreira model (GA-optimized) coefficients – INR . . . 167
A.138 De Rezende-Ferreira model (GA-optimized) coefficients – JPY . . . 168
A.139 De Rezende-Ferreira model (GA-optimized) coefficients – MYR . . . 168
A.140 De Rezende-Ferreira model (GA-optimized) coefficients – NZD . . . 169
A.141 De Rezende-Ferreira model (GA-optimized) coefficients – TWD . . . 169
A.142 De Rezende-Ferreira model (GA-optimized) coefficients – USD . . . 170
A.143 De Rezende-Ferreira model (PSO-optimized) coefficients – AUD . . . 170
A.144 De Rezende-Ferreira model (PSO-optimized) coefficients – CNY . . . 171
A.145 De Rezende-Ferreira model (PSO-optimized) coefficients – HKD . . . 171
A.146 De Rezende-Ferreira model (PSO-optimized) coefficients – INR . . . 172
A.147 De Rezende-Ferreira model (PSO-optimized) coefficients – JPY . . . 172
A.148 De Rezende-Ferreira model (PSO-optimized) coefficients – MYR . . . 173
A.149 De Rezende-Ferreira model (PSO-optimized) coefficients – NZD . . . 173
A.150 De Rezende-Ferreira model (PSO-optimized) coefficients – TWD . . . 174
A.151 De Rezende-Ferreira model (PSO-optimized) coefficients – USD . . . 174
A.152 De Rezende-Ferreira model (SA-optimized) coefficients – AUD . . . 175
A.153 De Rezende-Ferreira model (SA-optimized) coefficients – CNY . . . 175
A.154 De Rezende-Ferreira model (SA-optimized) coefficients – HKD . . . 176
A.155 De Rezende-Ferreira model (SA-optimized) coefficients – INR . . . 176
A.156 De Rezende-Ferreira model (SA-optimized) coefficients – JPY . . . 177
A.157 De Rezende-Ferreira model (SA-optimized) coefficients – MYR . . . 177
A.158 De Rezende-Ferreira model (SA-optimized) coefficients – NZD . . . 178

A.159 De Rezende-Ferreira model (SA-optimized) coefficients – TWD 178
A.160 De Rezende-Ferreira model (SA-optimized) coefficients – USD 179
A.161 Overall Average Cumulative R^2 Bounds Comparison 180
A.162 Cumulative R^2 bounds – AUD 180
A.163 Cumulative R^2 bounds – CNY 181
A.164 Cumulative R^2 bounds – HKD 181
A.165 Cumulative R^2 bounds – INR 181
A.166 Cumulative R^2 bounds – JPY 182
A.167 Cumulative R^2 bounds – MYR 182
A.168 Cumulative R^2 bounds – NZD 182
A.169 Cumulative R^2 bounds – TWD 183
A.170 Cumulative R^2 bounds – USD 183
A.171 De Rezende-Ferreira Model: Base vs. GA Optimized Performance – FRANCE 184
A.172 De Rezende-Ferreira Model: Base vs. PSO Optimized Performance – FRANCE 184
A.173 De Rezende-Ferreira Model: Base vs. SA Optimized Performance – FRANCE 185
A.174 De Rezende-Ferreira Model: Base vs. GA Optimized Performance – GERMANY 185
A.175 De Rezende-Ferreira Model: Base vs. PSO Optimized Performance – GERMANY 186
A.176 De Rezende-Ferreira Model: Base vs. SA Optimized Performance – GERMANY 186
A.177 De Rezende-Ferreira Model: Base vs. GA Optimized Performance – ITALY 187
A.178 De Rezende-Ferreira Model: Base vs. PSO Optimized Performance – ITALY 187
A.179 De Rezende-Ferreira Model: Base vs. SA Optimized Performance – ITALY 188
A.180 De Rezende-Ferreira Model: Base vs. GA Optimized Performance – SPAIN 188
A.181 De Rezende-Ferreira Model: Base vs. PSO Optimized Performance – SPAIN 189
A.182 De Rezende-Ferreira Model: Base vs. SA Optimized Performance – SPAIN 189
A.183 De Rezende-Ferreira Model: Base vs. GA Optimized Performance – UK 190
A.184 De Rezende-Ferreira Model: Base vs. PSO Optimized Performance – UK 190
A.185 De Rezende-Ferreira Model: Base vs. SA Optimized Performance – UK 191
A.186 Nelson-Siegel Model: Base vs. GA Optimized Performance – FRANCE 192
A.187 Nelson-Siegel Model: Base vs. PSO Optimized Performance – FRANCE 192

A.188 Nelson-Siegel Model: Base vs. SA Optimized Performance – FRANCE . . . 193
A.189 Nelson-Siegel Model: Base vs. GA Optimized Performance – GERMANY . . . 193
A.190 Nelson-Siegel Model: Base vs. PSO Optimized Performance – GERMANY . . . 194
A.191 Nelson-Siegel Model: Base vs. SA Optimized Performance – GERMANY . . . 194
A.192 Nelson-Siegel Model: Base vs. GA Optimized Performance – ITALY . . . 195
A.193 Nelson-Siegel Model: Base vs. PSO Optimized Performance – ITALY . . . 195
A.194 Nelson-Siegel Model: Base vs. SA Optimized Performance – ITALY . . . 196
A.195 Nelson-Siegel Model: Base vs. GA Optimized Performance – SPAIN . . . 196
A.196 Nelson-Siegel Model: Base vs. PSO Optimized Performance – SPAIN . . . 197
A.197 Nelson-Siegel Model: Base vs. SA Optimized Performance – SPAIN . . . 197
A.198 Nelson-Siegel Model: Base vs. GA Optimized Performance – UK . . . 198
A.199 Nelson-Siegel Model: Base vs. PSO Optimized Performance – UK . . . 198
A.200 Nelson-Siegel Model: Base vs. SA Optimized Performance – UK . . . 199
A.201 Svensson Model: Base vs. GA Optimized Performance – FRANCE . . . 200
A.202 Svensson Model: Base vs. PSO Optimized Performance – FRANCE . . . 200
A.203 Svensson Model: Base vs. SA Optimized Performance – FRANCE . . . 201
A.204 Svensson Model: Base vs. GA Optimized Performance – GERMANY . . . 201
A.205 Svensson Model: Base vs. PSO Optimized Performance – GERMANY . . . 202
A.206 Svensson Model: Base vs. SA Optimized Performance – GERMANY . . . 202
A.207 Svensson Model: Base vs. GA Optimized Performance – ITALY . . . 203
A.208 Svensson Model: Base vs. PSO Optimized Performance – ITALY . . . 203
A.209 Svensson Model: Base vs. SA Optimized Performance – ITALY . . . 204
A.210 Svensson Model: Base vs. GA Optimized Performance – SPAIN . . . 204
A.211 Svensson Model: Base vs. PSO Optimized Performance – SPAIN . . . 205

A.212 Svensson Model: Base vs. SA Optimized Performance – SPAIN . . . 205
A.213 Svensson Model: Base vs. GA Optimized Performance – UK . . . 206
A.214 Svensson Model: Base vs. PSO Optimized Performance – UK . . . 206
A.215 Svensson Model: Base vs. SA Optimized Performance – UK . . . 207
A.216 Cumulative Performance (GA Optimized) – FRANCE 208
A.217 Cumulative Performance (PSO Optimized) – FRANCE 208
A.218 Cumulative Performance (SA Optimized) – FRANCE 209
A.219 Cumulative Performance (GA Optimized) – GERMANY . . . 209
A.220 Cumulative Performance (PSO Optimized) – GERMANY . . . 210
A.221 Cumulative Performance (SA Optimized) – GERMANY . . . 210
A.222 Cumulative Performance (GA Optimized) – ITALY . 211
A.223 Cumulative Performance (PSO Optimized) – ITALY 211
A.224 Cumulative Performance (SA Optimized) – ITALY . 212
A.225 Cumulative Performance (GA Optimized) – SPAIN . 212
A.226 Cumulative Performance (PSO Optimized) – SPAIN 213
A.227 Cumulative Performance (SA Optimized) – SPAIN . 213
A.228 Cumulative Performance (GA Optimized) – UK . . . 214
A.229 Cumulative Performance (PSO Optimized) – UK . . 214
A.230 Cumulative Performance (SA Optimized) – UK . . . 215
A.231 Global Performance Histogram – All Algorithms . . 216
A.232 Global Performance Histogram – GA 216
A.233 Global Performance Histogram – PSO 217
A.234 Global Performance Histogram – SA 217
A.235 Vasicek Forecast – FRANCE 218
A.236 Vasicek Forecast – GERMANY 218
A.237 Vasicek Forecast – ITALY 219
A.238 Vasicek Forecast – SPAIN 219
A.239 Vasicek Forecast – UK 220

Chapter 3: A reliable and prudent forward-looking Value at Risk estimation applied to an Option Portfolio 221

3.1 Cumulative Returns of Selected Companies by Sector. 229

4.1 Correlation Matrix for Selected Companies 244

5.1 Exxon Mobil Volatility Estimation (Max/Min). . . . 245
5.2 Minimum Volatility Method Used for Exxon Mobil (XOM) Over Time. 246
5.3 Exxon Mobil Drift Estimation (Max/Min) 246
5.4 Minimum Drift Method Used for Exxon Mobil (XOM) Over Time 247

5.5 Exxon Mobil Dividend Estimation (Max/Min) 247
5.6 Minimum Dividend Method Used for Exxon Mobil (XOM) Over Time 247
5.7 Time Series of Portfolio VaR at 10%, 5%, and 1% Percentiles 249

List of Tables

Chapter 1: Demand Forecasting for a Manufacturing Company **1**

2.1 Classification of demand based on CV^2 components. 7

3.1 VHIT xlsx file (fictitious time series of products sold) 12
3.2 PV VHIT xlsx file (fictitious families of different products) 13

4.1 Classification distribution for Family Product 19
4.2 Classification distribution for Family Product Client 19

Chapter 2: Improving the Performance of Traditional Interest Rates Term Structure Models Using Nature-Inspired Optimization Algorithms **50**

4.1 Granulometry and securities used for the zero rates bootstrap (data collected as of February 1st, 2018) . . 78
4.2 Cumulative percentage of curves with $R^2 > 0.95$ and valid bounds by method and currency 80
4.3 Fitting statistics for different currencies using a Machine Learning approach 81
4.4 Configuration parameters for optimization algorithms 83
4.5 In-sample RMSE values for yield curve fitting across currencies and models 84
4.6 Out-of-sample RMSE values for yield curve fitting across currencies and models 84
4.7 Overall performance comparison: Parametric models with optimization algorithms vs. Gaussian Process Regression 85
4.8 Performance comparison: Success rate ($R^2 \geq 0.95$ + Parameters within bounds) 89

Chapter 3: A reliable and prudent forward-looking Value at Risk estimation applied to an Option Portfolio **221**

3.1 Stock Selection by Sector. 228
3.2 Correlation Matrix of Daily Stock Returns. 229
3.3 Analysis of Stock Daily Return Descriptive Statistics. 230
3.4 Option Types and Corresponding Underlying Stocks. 230

3.5 Pricing formulas for Standard Barrier Options depending on Strike (X) and Barrier (H). Puts follow symmetric rules by changing ϕ and η. 235
3.6 Portfolio Details: Maximum Disaggregation 237

CHAPTER 1

Demand Forecasting for a Manufacturing Company

Abstract

The first chapter of my PhD thesis addresses the demand forecasting problem for VHIT, the company sponsoring my doctoral research. The objective of the thesis is to predict the future value of the sales for the different products sold by the company. Forecasting in the manufacturing sector is considered particularly complex, as the time series of sold products are often intermittent and therefore difficult to forecast. These types of time series cannot be accurately predicted using traditional models from econometrics or machine learning but require forecasting methods specifically designed for intermittent time series. In our case, VHIT sells hundreds of different products, and the majority of them are characterized by intermittent behavior; however, there are also products with more linear time series. Therefore, what we did was to create a forecasting engine able to classify the different types of products and then select the most suitable forecasting model for each type. The results obtained show that, generally, products with intermittent time series are predicted with greater accuracy than more linear ones. This contradicts what is stated in the literature but is justified by the fact that in the case of VHIT, intermittent and linear time series are not equally distributed. In fact, we have a minority of non-intermittent time series that account for the vast majority of revenue, while we also have a large number of products with intermittent time series that, in terms of revenue, have little value. These products record very few sporadic orders or even none at all, making them easier to predict.

Keywords: Classification, Forecasting, Intermittent Time Series, Manufacturing, Automotive.

J.E.L. Classification: C53, C45, E17, L62, L23

Acknowledgements

I gratefully acknowledge the valuable comments and suggestions received during the presentation of this study at the EURO 2025 – European Conference on Operational Research (University of Leeds, United Kingdom, June 22–25, 2025)

I'm particularly thankful to the conference participants and anonymous reviewers for their constructive feedback, which has significantly improved the quality of this manuscript.

1 Introduction

It is a well-established fact in forecasting that no single model consistently outperforms all others across every scenario. Rather, a diverse array of methods exists, requiring the selection of the most appropriate model based on the specific context.

In the early 1970s, some researchers realized that forecasting in the manufacturing context posed a particularly complex challenge due to the intermittent nature of the time series being analyzed. In such cases, traditional econometric models often failed [Syntetos and Boylan 2005], requiring alternative approaches to be developed.

In this thesis, I aim to analyze the journey undertaken by researchers who have worked on intermittent demand forecasting through a literature review, starting from the earliest approaches and progressing towards the most effective models that today represent the state of the art in this field.

Subsequently, I will present the dataset provided by VHIT, on which this study is based. The data is confidential and thus will not be shown in its entirety; however, the data cleaning process will be thoroughly explained, showing how we arrived at the final time series of product sales upon which the forecasting models will be applied.

Before proceeding with forecasting, the crucial phase of classification will be described. Without this step, we would be unable to distinguish between different products based on their historical sales patterns. Classification is, in fact, a critical moment that precedes the actual forecasting and allows us to understand the past behavior of product sales by analyzing the Average Demand Interval (ADI) and the squared Coefficient of Variation (CV^2).

Once the time series are classified into four macro-categories (Erratic, Intermittent, Lumpy, and Smooth), we are ready to proceed with forecasting. The forecasting section presents the models used depending on the category determined by the classification. Specifically, the Teunter-Syntetos-Babai (TSB) model is applied to the Lumpy series. For all non-Lumpy series, the SARIMA model is used if the series is stationary, and the RNN (Recurrent Neural Network) model is applied if the series is non-stationary.

Finally, the results obtained from each model will be presented and discussed.

2 Literature Review

2.1 Classification

In the context of demand forecasting, as well as inventory management, it is common to face the challenge of managing a wide range of products with very different characteristics. As highlighted by [Eaves and Kingsman 2004], in their analysis of the Royal Air Force's demand data, which included thousands of different product codes, to accelerate the forecasting process, products are grouped into homogeneous families, and a different forecasting model is applied to each family, chosen based on the characteristics of the products in the family. This is done through classification methods that aim to identify similarities between products, grouping them based on one or more discriminatory variables.

The classification of sold products and inventory can be carried out in various ways. However, there is no universally accepted standard for such classification, and ad hoc categorizations are often adopted both in the academic literature and in industrial software applications [Babiloni et al. 2012].

One of the most recognized methods in inventory classification is the ABC model, based on Pareto's studies. This method categorizes products according to a relevant variable, which can be determined by different criteria. A traditional approach relies on the total sales value, calculated as price multiplied by quantity, as proposed by [Dhoka 2013]. Alternatively, other variables can be considered, such as obsolescence, lead times, substitutability, repairability, criticality, and commonality, as suggested by [Flores 1987]. These factors can be combined in various ways to refine the classification process.

Beyond the ABC model, several other classification methods exist, which can be applied either individually or in combination with ABC and other techniques. Among them:

- **XYZ Analysis** – Classifies products based on demand stability, distinguishing between items with stable, moderately variable, and highly unpredictable demand.
- **VED Analysis** – Organizes products according to operational criticality, identifying vital, essential, and desirable items for business functionality.
- **HML Analysis** – Categorizes products based on unit cost, differentiating between high, medium, and low-cost items.

- **FSN Analysis** – Evaluates products based on movement frequency, classifying them as fast-moving, slow-moving, or non-moving, as applied by [Hemeimat et al. 2016] before conducting the study on forecasting spare parts demand.

These methods, which have been widely used in warehouse management, have nevertheless proven to be ineffective as classification tools when applied in preparation for forecasting models based on selected product groups.

As noted in [Boylan and Syntetos 2008], the main limitation of the ABC model in demand forecasting is that it does not consider the shape of the product's historical time series. Classifying items based on their economic value or revenue can lead to grouping products with very different demand frequency and volume within the same category.

Indeed, if value is chosen as the discriminating variable, we might have two products in Group A, both relevant in terms of economic importance: one with a regular and consistent time series, and the other with a highly volatile and intermittent pattern. In this case, the forecasting model applied to both products would be ineffective for at least one of them.

If a model suited for smooth time series is chosen, the product with an intermittent demand will not be forecasted correctly. Conversely, if a model designed for intermittent time series is applied, the product with regular demand will be penalized. This happens simply because the significant differences in demand patterns are not taken into account.

2.1.1 Time series classification

Clearly, classification is a crucial step for product forecasting, as an inaccurate classification will lead to using unsuitable models for our products, and the predicted result will be imprecise [Boylan et al. 2008]. A process so delicate, on which the final outcome depends, cannot be managed through a simple classification model like the ABC method, which bases product diversification on attributes such as value or relevance. Rather, it is understood that since forecasting models draw directly from the time series of the products, and predictions are formulated based on these time series, product classification must be done by grouping products with similar time series together, and thus suitable to be predicted by the same forecasting model.

The first to propose a classification of demand based on the characteristics of time series was [Williams 1984], who developed a method to decompose the variance of Demand during lead time (DDLT) into three causal components: the variability of transactions, the variability of demand size, and the variability of lead time. The equation for the variance partitioning that he formulated is as follows:

$$CV_{DDLT}^2 = CV_n^2 + CV_x^2 + CV_L^2 \tag{2.1}$$

where:

- CV^2_{DDLT} is the coefficient of variation of demand during lead time;
- CV_n is the coefficient of variation of the number of transactions;
- CV_x is the coefficient of variation of the demand size;
- CV_L is the coefficient of variation of lead time.

This equation allows categorizing demand based on the variability levels of its components, and Williams used these categories to differentiate various types of demand such as "smooth," "irregular," or "high movement."

Subsequently, [Eaves and Kingsman 2004] extended Williams' approach by further refining the demand categories through a more detailed distinction between the variability of transactions, demand size, and lead time. They proposed a more granular subdivision, which allows for better identification of demand structures, integrating the variability of transactions at the lower quartiles and the variability of demand size and lead time at their medians.

In this way, they introduced further categories, such as "moderately intermittent" and "highly intermittent," which provide a more precise view of the nature of the demand and allow for selecting more appropriate forecasting models for each type of time series. This detailed classification is summarized in Table 2.1.

CV^2DDLT	**Categorization of Demand**			
CV_n^2/L	**CV_x^2/nL**	**CV_L^2**	**Williams**	**Eaves and Kingsman**
Low	Low		Smooth	Smooth
Low	High		Smooth	Irregular
High	Low		Slow moving	Slow moving
High	High	Low	Sporadic	Mildly intermittent
High	High	High	Sporadic	Highly intermittent

Table 2.1: Classification of demand based on CV^2 components.

Although Williams, Evans, and Kingsman laid the foundation for demand classification based on the chronological characteristics of products, their studies were applied to specific cases and limited situations. In contrast, [Syntetos and Boylan 2005] aimed to develop a more universal and applicable approach to any sector or context. Their objective was to create a replicable method for demand classification, independent of the peculiarities of the sector.

Syntetos and Boylan identified two key variables that influence the structure of time series: the variability of demand quantity, measured through the Coefficient of Variation (CV^2), and the variability of the time between orders, measured by the Average inter-Demand Interval (ADI),. These two factors define four distinct scenarios.

The first scenario occurs when both the CV^2 and ADI are high, defining a "lumpy" demand. This type of demand is characterized by sporadic and unpredictable orders with great variability in the quantities requested, making forecasting difficult.

In the second scenario, with a high ADI and low CV, the demand is classified as "intermittent." Here, orders are infrequent, but the quantities tend to be more stable, reducing uncertainty regarding demand variability.

The third scenario, with a high CV^2 and low ADI, describes an "erratic" demand, characterized by frequent orders but with significant fluctuations in quantities, complicating the forecasting process.

Finally, when both the CV^2 and ADI are low, the demand is classified as "smooth," meaning it is regular and easier to forecast due to reduced variability in both the frequency and the quantities of orders. As shown in Figure 2.1, the different demand time series illustrate various patterns.

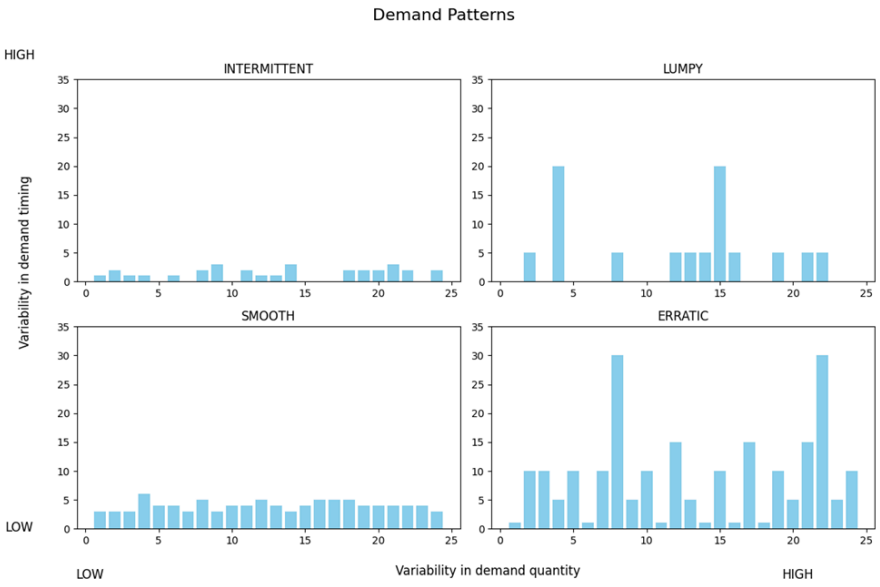


Figure 2.1: Demand time series

2.2 Intermittent Demand Forecasting

When we talk about intermittent demand, we refer to situations where the demand for a given product is particularly irregular and unpredictable. In these cases, many periods may pass without any orders, followed by a sudden request for quantities that are difficult to estimate in advance [**?**]. This combination of uncertain timing and variable order sizes makes it especially challenging to forecast future demand accurately. To address this issue, a dedicated line of research has emerged over the years, known as intermittent demand forecasting, which aims to develop forecasting tools better suited to these specific scenarios than traditional models.

When discussing forecasting methods for intermittent demand, the first significant contribution comes from [Croston 1972]. In his study, Croston addresses the problem of irregular demand, characterized by frequent periods without orders and sporadic orders of varying quantities. He proposed an innovative approach that separates the forecasting of demand quantity from the forecasting of the time interval between orders, overcoming the limitations of traditional methods such as moving averages and AutoRegressive Integrated Moving Average (ARIMA), which did not effectively handle periods without orders.

Croston's model is based on two main components: the estimate of the demand quantity $\hat{z}_t$ and the estimate of the average time interval between orders $\hat{p}_t$. The demand quantity is updated continuously according to the formula:

$$\hat{z}_t = \hat{z}_{t-1} + \alpha(z_t - \hat{z}_{t-1}), \tag{2.2}$$

while the interval between orders is estimated as:

$$\hat{p}_t = \hat{p}_{t-1} + \beta(p_t - \hat{p}_{t-1}), \tag{2.3}$$

where α and β are smoothing factors, typically between 0 and 1. Croston proposed to keep these factors constant, whereas [Schultz 1987] suggested making them variable to better adapt to the characteristics of the series. Once the estimated values of the demand and intervals are obtained, the forecasted demand for the next period can be calculated with the formula:

$$\hat{y}_{t+1} = \frac{\hat{z}_t}{\hat{p}_t}. \tag{2.4}$$

This approach allows for better management of intermittent series, although one of the main criticisms is that it tends to overestimate the average demand in the long run, especially when orders are rare. This limitation led to further improvements aimed at making the forecasts more accurate. [Syntetos and Boylan 2005] introduced a significant modification to Croston's method, known as the Syntetos-Boylan Approximation (SBA). Their work focuses on reducing the overestimation bias of the original method. The SBA formula introduces a correction factor that reduces the estimate of average demand, represented as:

$$\hat{d}_{\text{SBA}} = \hat{d}_{\text{Croston}} \times \left(1 - \frac{\alpha}{2}\right), \tag{2.5}$$

where α is the smoothing rate. This improvement makes SBA more accurate than Croston's method, especially for series with low order frequency. However, SBA maintains the assumption of stationarity and independence of demand, limiting its applicability in contexts with high variability or structural changes.

[Eaves and Kingsman 2004] conducted a study using demand data from spare parts inventory management. They compared various fore-

casting methods, including Croston's method, moving averages, exponential smoothing, and the Syntetos-Boylan Approximation (SBA). The results of their analysis confirmed the superiority of the SBA method over the others, highlighting that Syntetos-Boylan's approximation offers the best performance for reducing inventory while maintaining good service levels, making it the most effective forecasting method for managing intermittent demand.

The following year, the three scholars [Syntetos et al. 2005] collaborated to test and compare various intermittent demand forecasting methods, including Croston's method, the Syntetos-Boylan method (SBA), and Exponentially Weighted Moving Average (EWMA). The aim of the study was to develop a systematic categorization of demand patterns based on their characteristics, to select the most suitable forecasting method, overcoming the arbitrary approach commonly used in the industry. Using the Mean Squared Error (MSE) as a measure of accuracy, they compared the methods on 3000 intermittent demand series from the automotive industry. They introduced key coefficients for classification: the squared coefficient of variation (CV^2), which measures the variability in demand quantity, and the average interdemand interval (ADI), which represents the intermittency. The results demonstrated that Croston's method was superior when CV^2 was below 0.49 and ADI was less than 1.32, while the Syntetos-Boylan method proved better in other cases. This categorization allows for establishing rules to choose the most effective forecasting method based on demand characteristics. The work demonstrated that empirical comparisons of methods not only verify performance but also contribute to a more systematic understanding of demand patterns, providing a solid basis for model selection and making demand pattern categorization more robust and scientifically grounded.

In addition to the bias due to overestimating demand, already noted by Syntetos and Boylan and partially corrected by the SBA approximation, [Teunter et al. 2011] found that Croston's model is not able to adequately handle obsolescence. Specifically, the method fails to account for periods where no orders are received. In fact, Croston's model does not update its forecasts until new orders arrive, leaving the estimate unchanged during periods of inactivity.

According to the authors, this approach produces unrealistic results, as a product that has not been requested for a long time should see a decreasing probability of receiving new orders. To address these limitations, the three authors proposed the TSB method, which introduces a probabilistic component to better manage intermittent demand. Unlike Croston's method, the TSB continuously updates the probability of a positive demand in each period, even when no new orders are placed, gradually reducing it as the inactivity period lengthens.

$$V[p'_t] = p(1-p)\frac{b}{2-b}. \tag{2.6}$$

This formula shows how TSB manages demand variability more

realistically compared to Croston's method, dynamically adapting to the absence of orders and offering more accurate forecasts, especially for items subject to obsolescence.

3 Data

3.1 Description of the Dataset

To study the historical sales series of products sold by the company and to develop sales forecasting models, we started with two datasets provided by the company. The first dataset encompasses sales from the beginning of 2012 to the end of 2024 for all products As shown in Table 3.1. The key information relevant to our study includes the sale date, the quantity of items sold, and the product identification code, which allows us to trace back to the customer and the type of product in question. Although the dataset contains additional information, these three variables are the most pertinent for our analysis. Specifically, the product code enables us to categorize the sold products into homogeneous groups, the date allows us to track the sales history, and the quantity of items sold is the critical variable we aim to study for predicting future trends.

Month	**Product**	**Quantity**	**Price (€)**	**Day**
January	XZ09.AB23-245	270	13.62	02/01/2023
January	XZ09.BG10-PU	720	41.15	03/01/2023
January	YT07.QW35-U3X	240	11.87	04/01/2023
January	AB03.ZX41-UBW	897	21.23	05/01/2023
January	TR08.KL11-UBW	387	9.17	06/01/2023
January	FG02.WX29-UBW	528	8.52	07/01/2023
January	MN10.YT18-VT1	264	4.26	08/01/2023
January	MN10.YT18-VT2	528	8.52	09/01/2023
January	LP06.QR47-VT3	264	4.26	10/01/2023
January	KL55.XY99-VT4	32	0.47	11/01/2023
January	GH88.MN56-VT5	100	1.50	12/01/2023

Table 3.1: VHIT xlsx file (fictitious time series of products sold)

While we also have the selling price for each product, we deliberately chose not to consider this variable in our analyses. This decision is due to the selling price being influenced by numerous external factors, such as fluctuations in raw material costs and energy prices. As

previously explained, both are significant expense items for the company, and in some cases, variable components related to the price of raw materials are included in the final selling price in supply contracts. Additionally, other more impactful factors include inflation trends and managerial decisions. Inflation significantly distorts the value of current products compared to those sold over a decade ago. Moreover, it is common for supply contracts to include occasional discounts for individual clients, further complicating the use of price as a variable. Therefore, the price is excluded from our studied variables.

The second dataset provided, Table 3.2, contains information regarding the product family and sales status. The sold products have an identification code composed of 13 digits, such as F009.A07.235-245. These digits are divided into four blocks, providing various pieces of information. The type of product is indicated by the third block of three digits (235), and the customer to whom it is destined is indicated by the last block of digits (245). This information is crucial as it allows us to group the different products into their respective families, such as all products of a certain type (x) or all products destined for a specific client (y). Moreover, from this second dataset, we can determine whether the products in question are still being sold or are no longer part of the company's offerings. Specifically, if the column "Status" shows the value 40, the product is still sold by the company, while if "Status" is equal to 50, it means the product was previously sold but is no longer available.

P\N (13-digit)	**P/N (10-digit)**	**Product**	**Name**	**Status**
1234.567.890-123	1234567890	AX34-ZX12	Alpha	50
2345.678.901-234	2345678901	BY89-PL56	Beta	50
3456.789.012-345	3456789012	QW45-LM99	Gamma	50
4567.890.123-456	4567890123	RT78-KJ34	Delta	50
5678.901.234-567	5678901234	TY12-BV66	Epsilon	50
6789.012.345-678	6789012345	OP90-XC23	Zeta	40
7890.123.456-789	7890123456	VW56-NM11	Eta	50
8901.234.567-890	8901234567	KL99-YU88	Theta	40
9012.345.678-901	9012345678	MN23-TR45	Iota	40
0123.456.789-012	0123456789	ZX89-QW22	Kappa	50

Table 3.2: PV VHIT xlsx file (fictitious families of different products)

3.2 Data Cleaning Process

In order to work on the company's data, we first needed to carry out several data cleaning operations explained in Figure 3.1. In our dataset, sales are recorded when the company receives the payment from the customers, so the sale is linked to the day of payment. Given that sales do not occur frequently enough to cover all days, there are many days where sales = 0. The presence of zeroes is a problem when it comes to forecasting, so we need to minimize the number of zeroes as much as possible. First, we aggregated the daily data into monthly data. By using monthly data, the months with sales = 0 were significantly reduced without losing any particularly relevant information, since the sale was recorded on a specific day, but this does not mean the company only produced those components on that day.

Next, we ensured that the quantity sold per month was always greater than or equal to 0. While this may seem obvious, negative values for sold quantities can appear due to adjustments in the company's accounting system. In such cases, the negative value will be distributed over the previous positive values found in earlier months.

Subsequently, the second task is to merge the time series of products belonging to the same family. Although individual product codes are unique, some products belong to the same macro-family. Thanks to the identification codes, we can determine both the product category and the customer to whom it is sold. This allows us to provide the company with an analysis that considers both the macro trends of each product and the particularities of each customer.

We have merged the time series of different products based on their affiliation. First, we combined all products with the same *Product Name* and same *Product* category to obtain the sales time series for each product category. This group of time series will be referred to as "Family Product" from now on.

Then, to provide a more detailed analysis that accounts for past demand for each customer, we added another level of discrimination based on the end customer, which can be identified by the last 3 digits of the customer code. Therefore, we created another group of time series, this time filtering not only by *Product* and *Product Number* but also by the last 3 digits to capture the demand for each product related to different customers. This second group of time series will be called "Family Product Client" from now on.

If these conditions are met, we can merge the time series of multiple products, as they are effectively the same product sold to the same customer, which has had its name changed for inventory reasons. For forecasting purposes, it is appropriate to consider them as a single product.

Once the products have been grouped into families, we are ready for the final step: removing products that are no longer being sold. To determine which products are still for sale and which are not, we need to check the "Status" column. If the product is identified as "40,"

it means it is still part of the products being sold. If it is identified as "50," it is no longer on sale. In the case of dealing with not just a single product but multiple products unified into a family, a single code identified as "40" is enough to retain the entire family. Otherwise, if none of the codes that make up the family is identified as "40," it means that it is a discontinued product line for which a predictive analysis is no longer necessary.

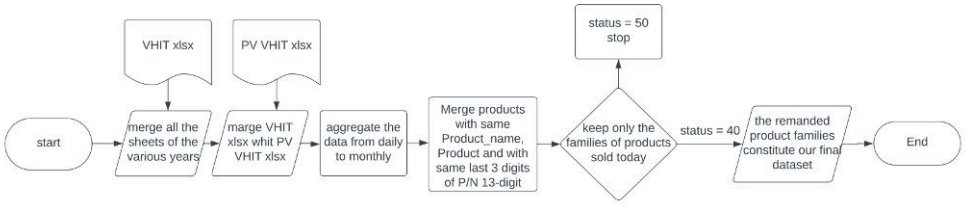


Figure 3.1: Cleaning Process

The data cleaning process described above, along with all subsequent steps undertaken to achieve the final forecasting of the time series, is extensively detailed in Appendix 8. This section provides a chronological walkthrough of the technical implementation carried out using Python code.

3.3 Preparing Data for Analysis

After these steps, we finally have the final dataset containing the product families on which we want to apply our forecasting analyses. In total, we ended up with two distinct groupings: 259 Family Products and 618 Family Product Clients. Initially, we had many individual products that, after the various steps previously discussed, were reduced and grouped into these families.

As we can see from the Pareto analyses below 3.2, 3.3, even though there are many families in both groupings, the company's revenue is supported by only a few key families, while the majority of the others are less significant in terms of generated revenue. This concentration of revenue in a small number of product families is observed in both the Family Product and Family Product Client categorizations.

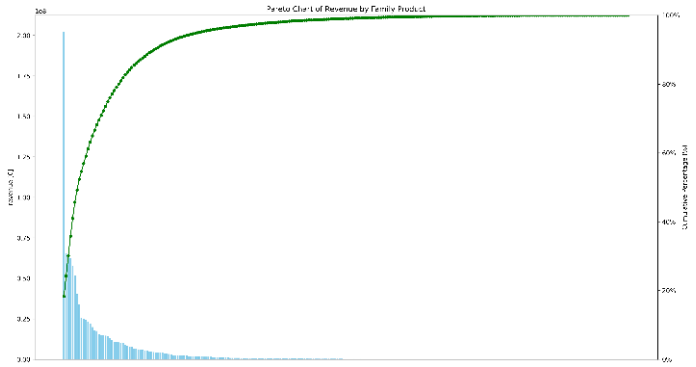


Figure 3.2: Pareto analysis of Family Products

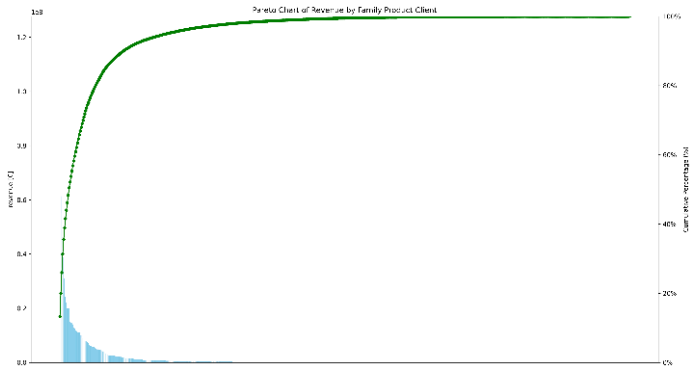


Figure 3.3: Pareto analysis of Family Product Clients

4 Classification

4.1 Time Series Classification Methodology

Once the data manipulation process was completed and the definitive time series were obtained, which serve as the foundation for applying various forecasting models, it becomes essential to classify these different families based on the characteristics of each time series. This classification process is fundamental as there exist different types of time series, and no single forecasting model is superior to all others in accurately predicting every type of time series.

Rather, there are various forecasting models, each with its own specificities, some suitable for predicting certain types of time series and others skilled at predicting others. Therefore, the necessary approach is to classify the different products within Family Product and Family Product Client categories to understand what types of time series are present and consequently apply the most performant forecasting model to each type, considering its unique characteristics.

The classification methodology is based on the framework proposed by Syntetos and Boylan, which utilizes two key metrics:

- **ADI (Average Demand Interval)**: Measures the demand frequency, distinguishing between regular and intermittent demand.
  - Threshold: Regular: ADI < 1.32
  - Intermittent: ADI ≥ 1.32
- **CV^2 (Coefficient of Variation squared)**: Indicates demand variability, distinguishing between stable and unstable demand.
  - Threshold: Stable: $CV^2 < 0.49$
  - Unstable: $CV^2 \geq 0.49$

Based on these indicators, time series can be classified into four categories:

- **Smooth**: Regular and stable demand
- **Intermittent**: Less frequent but stable demand
- **Erratic**: Regular but variable demand
- **Lumpy**: Both irregular and variable demand

This classification framework provides a systematic approach to identifying the appropriate forecasting method for each product family, optimizing prediction accuracy based on the inherent characteristics of the underlying time series.

4.2 Classification Results

As can be observed from the classification results in Figure 4.1, 4.2 and Table 4.1, 4.2, a significant portion of the families belongs to the Lumpy category, This is not surprising, given that in the manufacturing sector, intermittent time series are not an exception, but the norm. In fact, Croston's original interest in this problem stemmed precisely from the need to address an issue arising from the industrial world in general, and the manufacturing industry in particular.

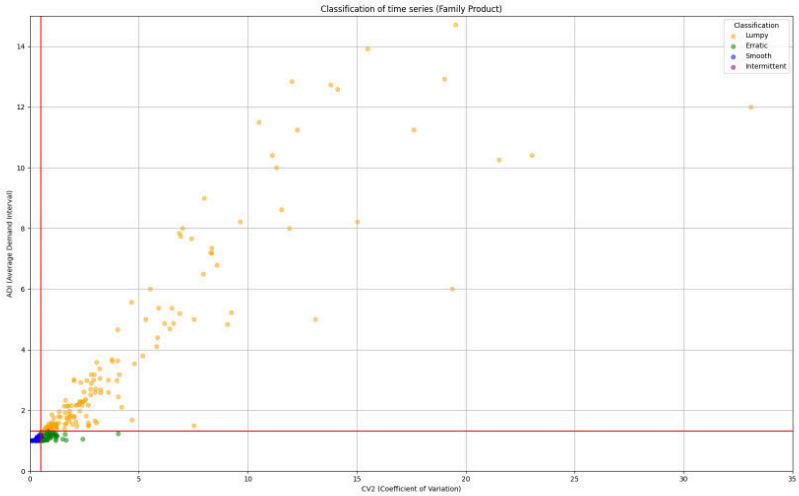


Figure 4.1: Classification of time series for Family Product

Classification	Total Families	Percentage of Families
Erratic	51	19.69%
Lumpy	169	65.25%
Smooth	39	15.06%

Table 4.1: Classification distribution for Family Product

In the case of Family Product Client, the presence of Lumpy series is further accentuated, which is normal because in this case, a greater number of time series is being observed in a more detailed manner, increasing the presence of situations characterized by intermittency.

Classification	Total Families	Percentage of Families
Erratic	47	7.61%
Intermittent	1	0.16%
Lumpy	515	83.33%
Smooth	55	8.90%

Table 4.2: Classification distribution for Family Product Client

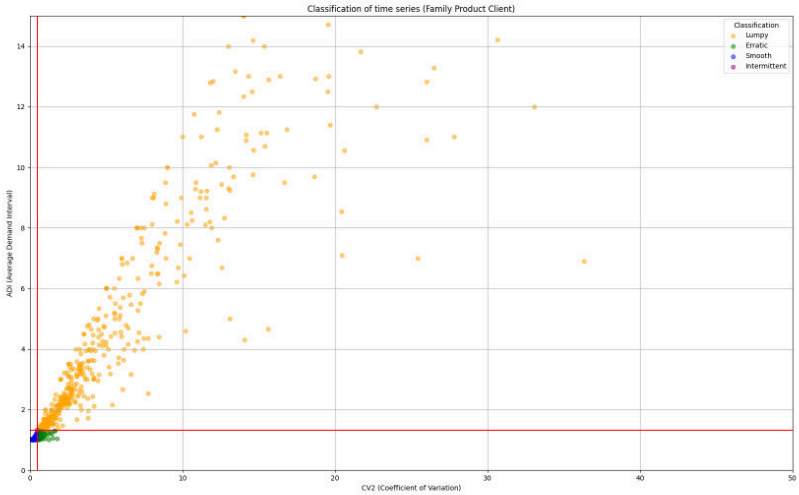


Figure 4.2: Classification of time series for Family Product Client

This was less evident in the series belonging to the Family Product category because, in order to obtain the sales time series for each product, multiple series belonging to different clients were aggregated, which helped eliminate some situations characterized by intermittency.

The predominance of Lumpy series, while entirely expected given the industrial context, is not good news, as Lumpy series remain among the different categories of time series the most difficult to predict since they do not present trends or recurring patterns. Observations are often irregular, and therefore these series are particularly unpredictable. However, the good news is that although in numerical terms these types of series are numerous, in terms of revenue generated for the company, they represent a minority share. As seen in the Pareto graphs 4.3 and 4.4, most of the revenue is developed by a few product families, while there are many different products that generate only a residual part of the turnover. Analyzing the Pareto analysis considering the different types of time series, it can be seen that Lumpy series are mostly situated in the tail and represent a negligible part of the revenue.

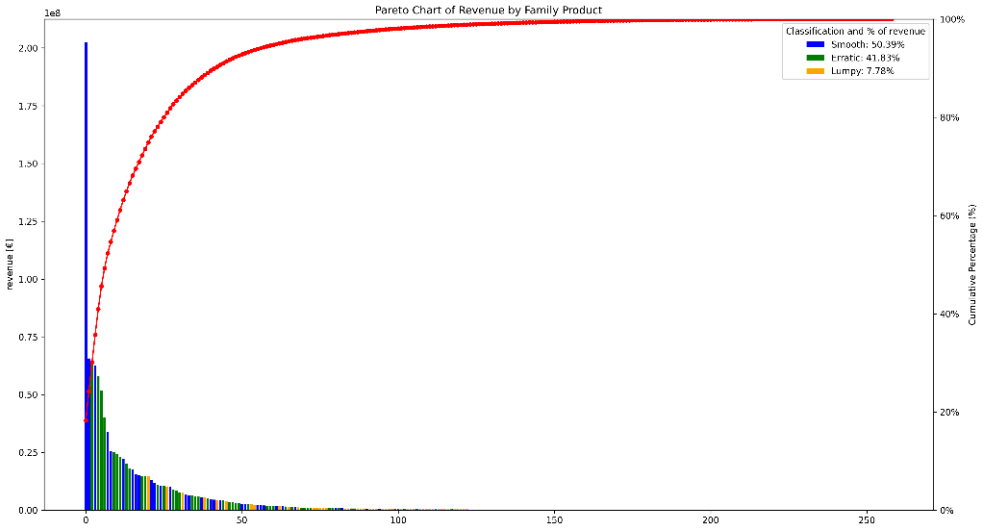


Figure 4.3: Pareto analysis by classification for Family Product

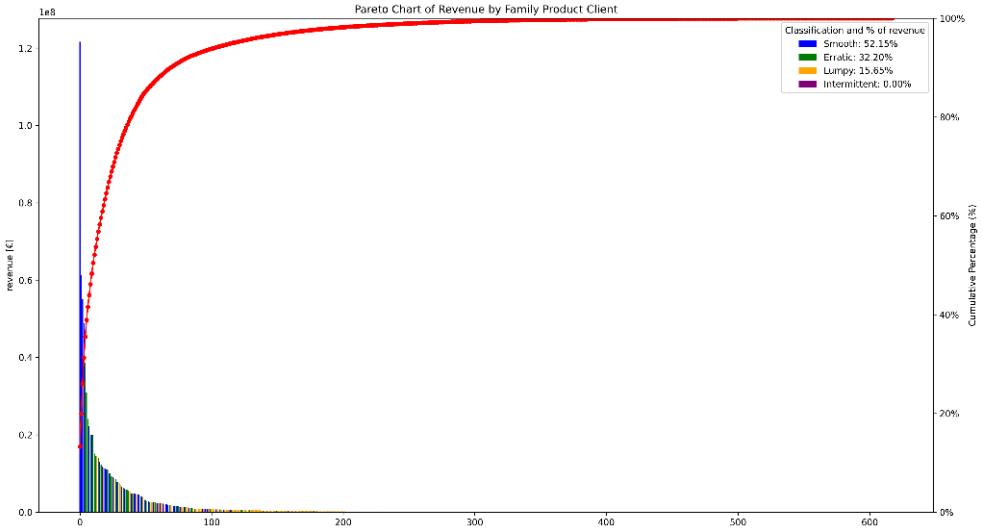


Figure 4.4: Pareto analysis by classification for Family Product Client

4.3 Revenue Impact Analysis by Classification

The analysis of revenue contribution by time series classification provides valuable insights into the business significance of each category. While Lumpy series dominate numerically as we can see in the Figures 4.6, 4.8, but their economic impact tells a different story as we can see in the Figures 4.5, 4.7.

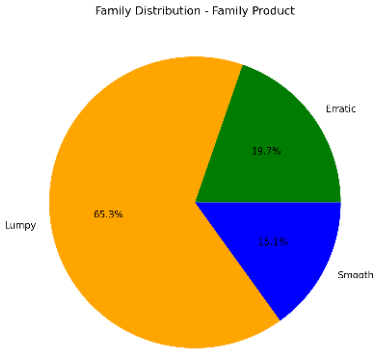


Figure 4.5: Count share by classification for Family Product

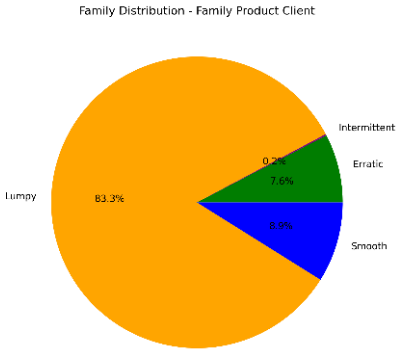


Figure 4.6: Count share by classification for Family Product Client

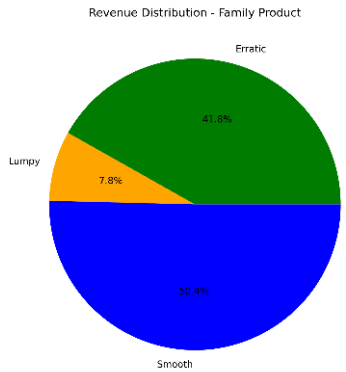


Figure 4.7: Revenue share by classification for Family Product

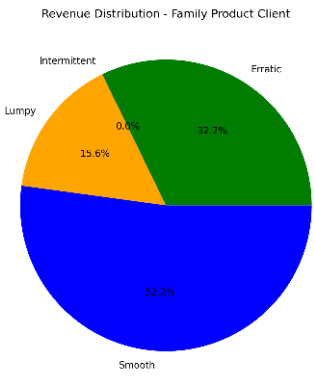


Figure 4.8: Revenue share by classification for Family Product Client

5 Forecasting

The classification process is not merely an end in itself but rather an intermediate step with the objective of identifying the behavior of time series. This process is a necessary preliminary step to reach at the selection of which forecasting method to apply to the different time series.

Considering the variety of time series patterns exhibited by products sold by VHIT, multiple forecasting models will be required. For Lumpy time series, which are characterized by intermittent demand patterns, specialized models designed for intermittent time series will be necessary. For other types of time series that present less erratic patterns, more conventional models such as SARIMA (Seasonal Autoregressive Integrated Moving Average) and RNN (Recurrent Neural Networks) can be employed, depending on the presence or absence of stationarity in the data. Below in figure 5.1 we can see the flow chart explaining the sorting of classified products

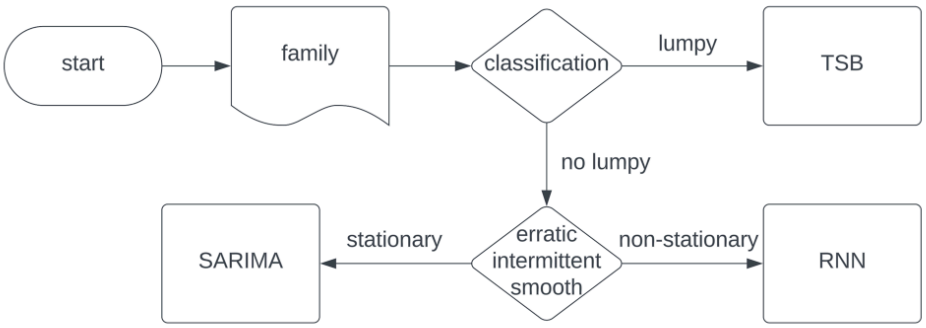


Figure 5.1: Overview of the forecasting methodology selection process

5.1 Forecasting Intermittent Time Series

For *lumpy* time series, characterized by high values of the Average Demand Interval (ADI) and Coefficient of Variation squared (CV^2), and thus by greater unpredictability in demand, the TSB (Teunter, Syntetos & Babai) model is adopted. This model was developed as a correction to the biases present in Croston's method, which, despite being an established approach for forecasting intermittent series, introduces systematic distortions in demand estimation.

Croston's method is based on the hypothesis that positive demand and intervals between demands follow independent stochastic processes. The forecast update occurs separately for the size of positive demand z_t and for the length of the interval between two demand events p_t, according to the following exponential smoothing equations:

$$z_t = \alpha d_t + (1 - \alpha) z_{t-1} \tag{5.1}$$

$$p_t = \alpha I_t + (1 - \alpha) p_{t-1} \tag{5.2}$$

where d_t represents the value of the observed demand in period t (non-zero), and I_t is the time elapsed since the last positive demand. The smoothing parameter α (with $0 < \alpha < 1$) determines the weight attributed to new observations relative to historical data. The forecast of future demand is therefore given by the ratio between the average size of positive demand and the average interval between two demand events:

$$\hat{d_t} = \frac{z_t}{p_t} \tag{5.3}$$

However, Croston's method presents two main biases:

1. **Overestimation of demand** – Since updating occurs only in periods where positive demand is observed, the model tends to underestimate the average length of intervals, and consequently overestimate the forecasted demand.

2. **Erroneous assumption about the distribution of intervals** – The model implicitly assumes that intervals between demands are geometrically distributed, an hypothesis often not valid in real data.

To correct these issues, the TSB method introduces a new component, namely the estimated probability θ_t that a period contains a positive demand. The update of this probability follows the rule:

$$\theta_t = \beta y_t + (1 - \beta) \theta_{t-1} \tag{5.4}$$

where y_t is an indicator variable that takes value 1 if the demand in t is positive, and 0 otherwise. The smoothing parameter β (with

$0 < \beta < 1$) regulates the degree of adaptation of the probability with respect to past observations.

The average size of positive demand continues to be updated as in Croston:

$$z_t = \alpha d_t + (1 - \alpha) z_{t-1}$$

However, instead of calculating the forecast as the ratio $\frac{z_t}{p_t}$, the TSB method corrects the demand overestimation bias by introducing the following expression:

$$\hat{d}_t = z_t \cdot \theta_t \tag{5.5}$$

This formulation ensures that the forecast is appropriately scaled according to the probability of observing positive demand, thus reducing the systematic error introduced by Croston's method.

In our case, both alpha (α) and beta (β) parameters were set to 0.8, as it was observed that historically, changes in sales trends occur very rapidly, both for products that were previously selling poorly and then begin to sell well, and for products that were once selling well but then experience a sharp decline.

The TSB method produces a forecast that initially appears as a flat, constant line over time. However, this forecast is progressively recalibrated as time passes, both in the presence of new demand and in its absence.

When a new order occurs, the model updates both the average size of positive demand z_t and the estimated probability θ_t, thus determining a revision of the future forecast. Conversely, in the absence of demand for an extended period, the value of θ_t gradually decreases due to the damping term $(1 - \beta)\theta_{t-1}$, resulting in a lowering of the future demand forecast.

Below we can see the prediction obtained by applying the TSB model to a lumpy product 5.2

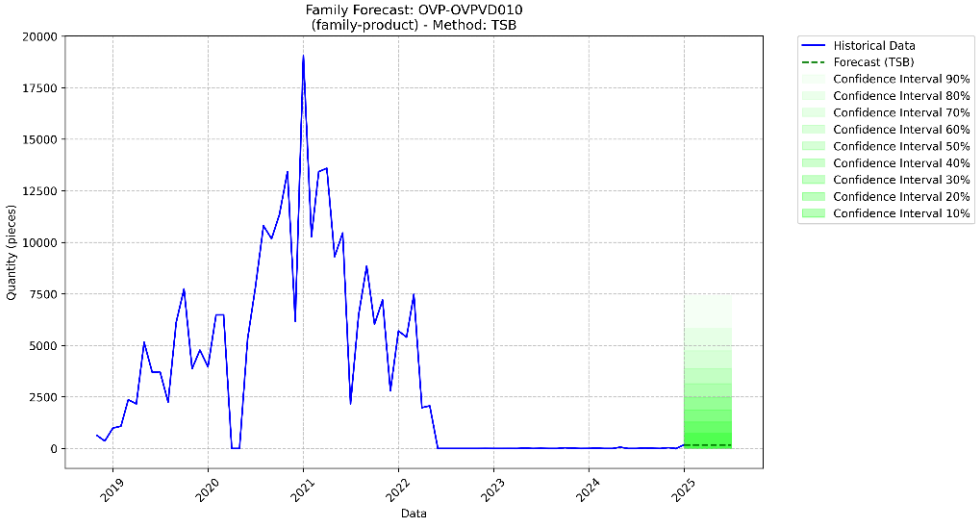


Figure 5.2: Example of TSB model forecast adjustment over time

5.2 Forecasting Stationary Time Series

As observed, both the **TSB** (Teunter-Syntetos-Babai) and **Croston** methods are not capable of providing accurate forecasts but are limited to estimating the probability of the next order. This limitation stems less from model inadequacies than from the inherent nature of the data itself. In the presence of sporadic, occasional, and irregular observations, there are currently no approaches that can guarantee better predictions. However, it is important to emphasize that the remaining product families, which do not exhibit "lumpy" characteristics, although numerically limited, constitute the majority of the revenue generated by the company. These product families, belonging to the **erratic**, **intermittent**, and **smooth** categories, can be predicted using more sophisticated models, which are not limited to suggesting the most probable quantity that will be requested in the future but attempt to provide a precise estimate of this quantity.

To address this challenge, we have decided to use two of the most established and widely applied forecasting models in time series analysis: the **SARIMA** model, from the field of econometrics, and the **RNN** (Recurrent Neural Network) model, which falls within the broader context of machine learning. The choice of these two models is not coincidental; in addition to their proven reliability, they are well-suited for integration within a unified framework. The **SARIMA** model requires that the analyzed time series be **stationary**, meaning that their fundamental characteristics, such as mean and variance, remain constant over time. In contrast, the **RNN** model does not impose this restriction, as machine learning allows capturing complex and non-linear relationships between data, even in the presence of non-stationary time series.

Therefore, the approach we will adopt is as follows: if the historical series of the analyzed products are stationary, these products will be forecast using the **SARIMA** model; in the case where the series are not

stationary, we will apply the **RNN** model. Determining the stationarity of the historical series is, therefore, a fundamental step. This can be done through the **Dickey-Fuller test**, which allows verifying the presence of a unit root in the time series and, thus, determining its stationarity.

The general formula for the Dickey-Fuller test is as follows:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \varepsilon_t \tag{5.6}$$

where:

- Δy_t: represents the first difference of the series at time t (calculated as $y_t - y_{t-1}$);
- α: is the intercept (drift) constant;
- β: is the coefficient of the deterministic time trend t;
- γ: is the coefficient of the lagged variable y_{t-1} ;
- ε_t: represents the white noise error term.

The Dickey-Fuller test verifies the null hypothesis that the series has a unit root (indicating non-stationarity). If the p-value associated with the test is less than the chosen significance level (0.05), the null hypothesis can be rejected, and it can be concluded that the series is stationary.

Once the stationarity of the series is determined, we will proceed with **forecasting**, using the most appropriate model for each product, based on the classification obtained regarding the stationarity of the time series. If the series is stationary, we will apply **SARIMA**, while for non-stationary series, we will proceed with the use of **RNN**. In this way, we adopt a targeted approach that leverages the potential of traditional models and advanced machine learning techniques.

Stationary time series are predicted using the SARIMA (Seasonal Autoregressive Integrated Moving Average) model, which extends ARIMA models by including a seasonal component. The general formulation of the model is as follows:

$$\Phi_p(B)\phi_P(B^s)(1-B)^d(1-B^s)^D y_t = \Theta_q(B)\theta_Q(B^s)\varepsilon_t \tag{5.7}$$

where the seasonal component is introduced through the polynomials $\phi_P(B^s)$ and $\theta_Q(B^s)$, which allow capturing periodic dependencies in the series. However, in our specific case, this component rarely appears in the analyzed time series, making its impact marginal in the modeling.

SARIMA describes the time series through autoregressive (AR) components of order p, which model the dependency on past values, and moving average (MA) components of order q, which represent the effect of previous errors. The I (integration) component, which normally manages the differentiation of the series to make it stationary, is not considered in our case, as we adopt the SARIMA model exclusively

for already stationary series, while for non-stationary series we use RNN-type models.

The identification of optimal parameters p, d, q for the non-seasonal part and P, D, Q, s for the seasonal part is carried out through selection criteria such as AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion), which allow balancing the goodness of fit of the model and its complexity, avoiding the risk of overfitting.

The AIC and BIC criteria are defined as:

$$\text{AIC} = 2k - 2\ln(L) \tag{5.8}$$

$$\text{BIC} = k\ln(n) - 2\ln(L) \tag{5.9}$$

where k is the number of parameters in the model, n is the sample size, and L is the likelihood of the model. The model with the lowest AIC or BIC value is typically selected as the optimal one.

Once the parameters are estimated, the SARIMA model can generate forecasts for a horizon h using the relationship:

$$y_{t+h} = \alpha_0 + \sum_{i=1}^{p} \phi_i y_{t+h-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t+h-i} \tag{5.10}$$

which allows determining the future value of the series by combining the influence of past values and previous errors, based on the dependencies learned by the model.

Below we can see the prediction obtained by applying the SARIMA model to a stationarity and not lumpy product 5.3

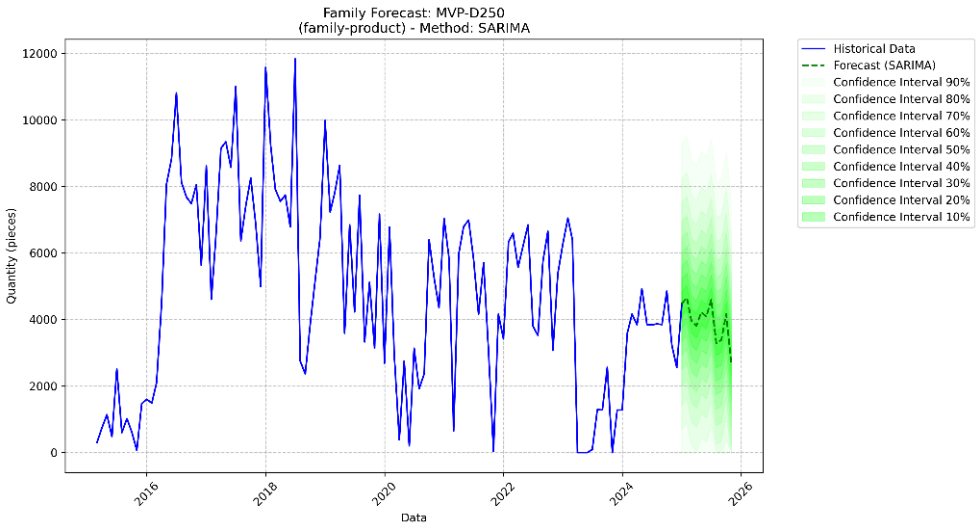


Figure 5.3: Example of SARIMA model forecast adjustment over time

5.3 Forecasting Non-Stationary Time Series

For modeling non-stationary time series, an architecture based on Recurrent Neural Networks (RNN) was adopted. The choice of this approach, rather than using more complex autoregressive deep learning models like Long Short-Term Memory (LSTM), is motivated by the limited availability of data. Our dataset covers the period from the beginning of 2012 to the end of 2024, with observations aggregated at a monthly level. Consequently, the maximum number of data points available for each time series amounts to $n = 12 \times (2024 - 2012 + 1) = 156$ observations. This quantity is suboptimal for training LSTM models, which generally require a greater amount of data for proper generalization. Therefore, an RNN structure was chosen, which is more suitable for contexts with limited datasets.

Recurrent Neural Networks process sequential data by maintaining an internal memory of previous inputs through recurrent connections. At each time step t, the network receives an input x_t and the previous hidden state h_{t-1}, producing a new hidden state h_t and an output y_t according to:

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b_h) \tag{5.11}$$

$$y_t = W_y h_t + b_y \tag{5.12}$$

where $W_h \in \mathbb{R}^{d_h \times d_h}$ represents the weight matrix for the recurrent connections, $W_x \in \mathbb{R}^{d_h \times d_x}$ is the weight matrix for the input transformation, $W_y \in \mathbb{R}^{d_y \times d_h}$ is the output weight matrix, and b_h, b_y are the corresponding bias vectors. The hyperbolic tangent activation function introduces non-linearity into the model. The recurrent structure can be visualized by unrolling the network through time as a chain of identical modules:

$$x_1 \xrightarrow{\text{RNN}} h_1 \xrightarrow{\text{RNN}} h_2 \xrightarrow{\text{RNN}} \cdots \xrightarrow{\text{RNN}} h_T \tag{5.13}$$

During training, the network learns optimal weight values by minimizing the Mean Squared Error:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{5.14}$$

where N is the number of training samples, y_i represents actual values, and $\hat{y}_i$ denotes predicted values. The optimization was performed using the Adam optimizer. Gradient computation is performed through Backpropagation Through Time (BPTT), where the gradient at time t depends on both current output error and error propagated from future time steps:

$$\frac{\partial \mathcal{L}}{\partial h_t} = \frac{\partial \mathcal{L}}{\partial y_t} \frac{\partial y_t}{\partial h_t} + \frac{\partial \mathcal{L}}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_t} \tag{5.15}$$

Dropout regularization was incorporated to mitigate overfitting.

For model training, the available data were divided into two sets: 80% of the time series was used for the training phase, while the remaining 20% was reserved for validation. In the learning process, a

sliding time window with a width of 12 observations (equivalent to one year of data) was adopted to capture seasonal patterns. Given a time series $\{x_1, x_2, \ldots, x_n\}$, this approach creates training samples:

$$\{(x_i, x_{i+1}, \ldots, x_{i+11}), x_{i+12}\}_{i=1}^{n-12} \tag{5.16}$$

where each sequence of 12 consecutive observations is used to predict the subsequent value. The model was trained exploring different configurations characterized by different hyperparameters: number of neurons in the hidden layer (d_h), dropout rate ($p_{\text{drop}} \in [0, 1]$), number of epochs, and batch size. Once the training phase was completed, predictions were compared with actual values of the validation portion to select the configuration that minimized the Root Mean Square Error:

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{val}}} \sum_{i=1}^{N_{\text{val}}} (y_i - \hat{y}_i)^2} \tag{5.17}$$

After identifying the optimal configuration, the model was retrained using the entire available dataset, i.e., the union of the training and validation data. Subsequently, the model was used to generate predictions for future time horizons. The length of the prediction horizon was determined as a function of the length of the time series, setting it equal to 10% of the length of the series itself. This choice was dictated by the variability in data availability: although some time series originated in 2012, others were introduced more recently. The adoption of a criterion based on a percentage rather than an absolute value ensured a more flexible adaptation to the different lengths of the series, maintaining a prediction horizon proportionate to the quantity of observed data.

Below we can see the prediction obtained by applying the RNN model to a non-stationarity and not lumpy product 5.4

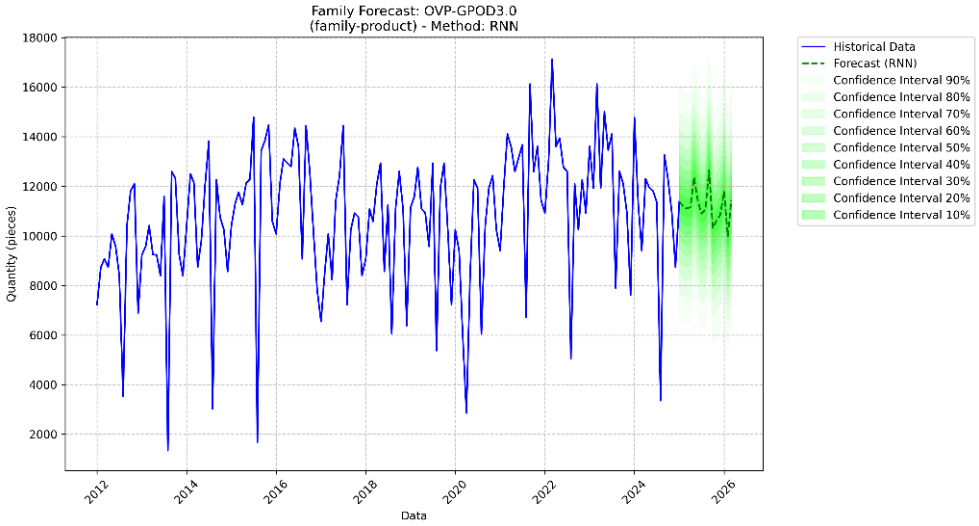


Figure 5.4: Example of RNN model forecast adjustment over time

6 Results

The analysis was initially conducted on the first dataset provided, covering the period from 2012 to 2023. Given that the data was aggregated at a monthly level, the resulting time series were limited in length. Consequently, to maximize the information available, we decided to utilize the entire dataset for the initial forecasting phase, omitting the traditional split into training and testing sets, as this would have excessively reduced the number of available observations. Subsequently, upon receiving the actual sales data for 2024 from the company, we validated our projections against these recent figures. Finally, after calibrating the models based on this comparison, a forecast for the year 2025 was generated.

6.1 Performance of the TSB model

The TSB model, although conceptually simple and based on a probabilistic logic of demand, has shown rather reliable results. This is due to the fact that most of the analyzed products have a reduced number of observations or even prolonged absences of demand.

Setting high values for the parameters α and β (equal to 0.8) allowed the model to adapt quickly to new observed values, ensuring consistent predictions in contexts characterized by static demand. However, this same characteristic translates into high vulnerability in cases of sudden trend reversal, such as a strong increase or decrease in orders. This leads to initially inaccurate predictions for such products, although, overall, the results obtained for the entire dataset are satisfactory.

In the specific case, products classified as "lumpy", i.e., characterized by long periods of absence of demand followed by irregular and concentrated orders, represent a minority compared to other products, which instead show greater stability. This contributed to the good performance of the TSB model.

Below in figure 6.1 we can see in green the forecast assumed with the confidence bands and in red the actual value of pieces sold

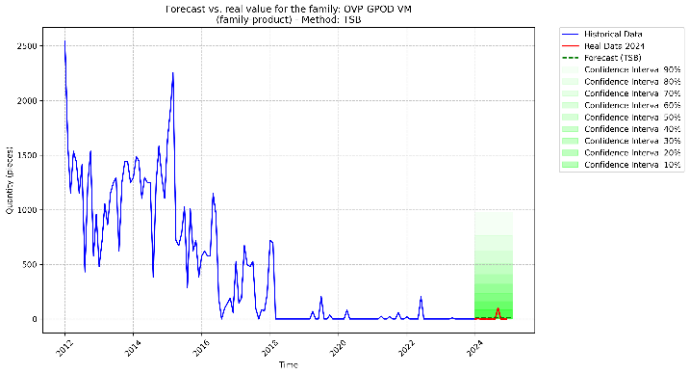


Figure 6.1: TSB model validation example - Family Product

The heatmap in the figure 6.2 shows the average percentage of correctly predicted products using the TSB model, across different forecast horizons (in months) and confidence levels.

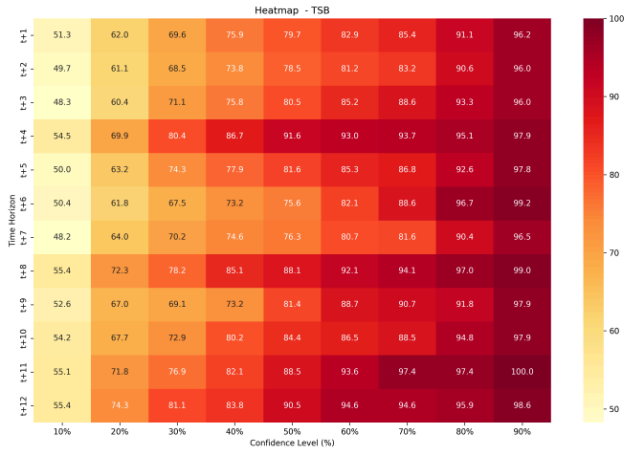


Figure 6.2: Heatmap of forecasting errors for the TSB model

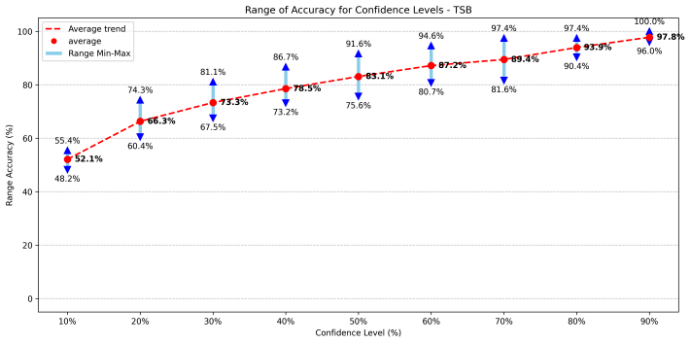


Figure 6.3: Level of accuracy of predictions for the TSB model

Figure 6.3 shows that to obtain a reliable forecast for 75% of the observations in the considered period, it is necessary to apply a confidence band equal to 30%.

6.2 Performance of the SARIMA model

Unlike the TSB model, the SARIMA model has shown less satisfactory results. The considered time series, while not completely "lumpy", present high volatility. In particular, the products under analysis show continuous demand over time, but with fluctuating trends, making it difficult to obtain accurate predictions.

The main difficulty in the SARIMA model is related to the nature of the time series: while ARIMA and SARIMA models are generally effective in capturing trends and seasonal components, the presence of irregular and highly non linear fluctuations in the analyzed data has compromised the accuracy of the predictions.

Below in figure 6.4 we can see in green the forecast assumed with the confidence bands and in red the actual value of pieces sold.

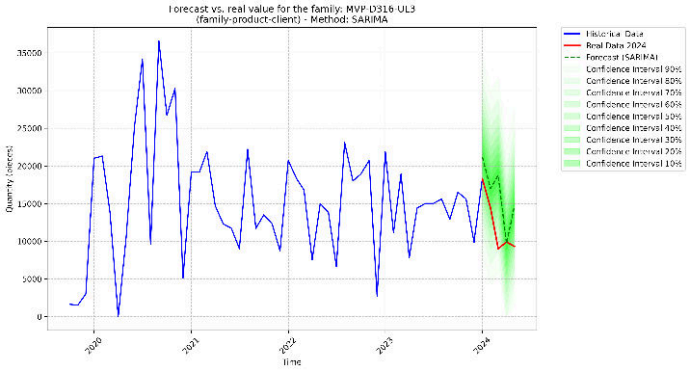


Figure 6.4: SARIMA model validation example - Family Product

The heatmap in the figure 6.5 shows the average percentage of correctly predicted products using the SARIMA model, across different forecast horizons (in months) and confidence levels.

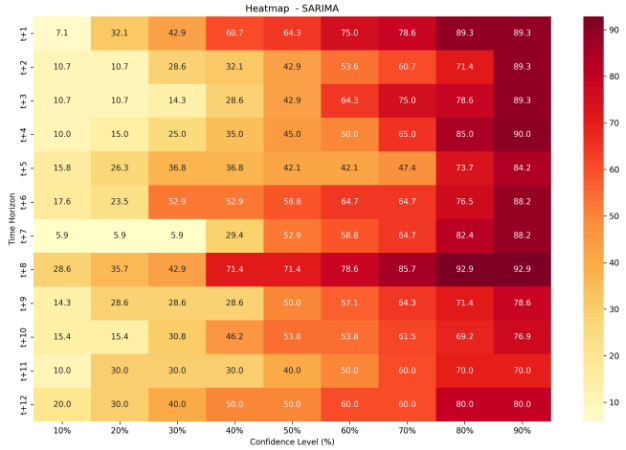


Figure 6.5: Heatmap of forecasting errors for the SARIMA model

Figure 6.6 shows that to ensure an acceptable margin of reliability in the predictions for 75% of the observations, a confidence band equal to 80% was necessary.

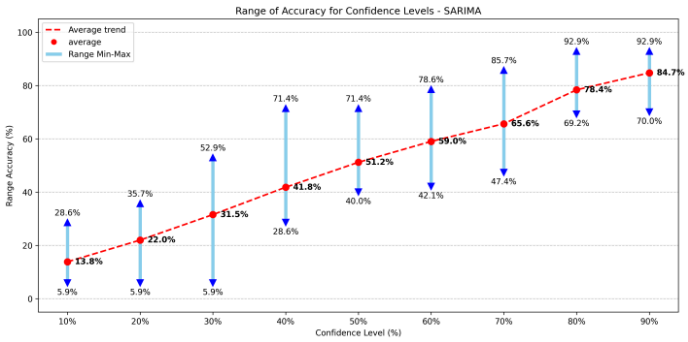


Figure 6.6: level of accuracy of predictions for the SARIMA model

6.3 Performance of the RNN model

Finally, the RNN model has shown performance comparable to that of the SARIMA model, albeit with slightly more encouraging results.

The neural network architecture allowed better modeling of the complexity of the time series compared to SARIMA, reducing the mean forecasting error. However, the absence of a large amount of historical data limited the network's ability to learn complex patterns, still leading to inferior results compared to more sophisticated models in contexts with more extensive datasets.

Below in Figure 6.7 we can see in green the forecast assumed with the confidence bands and in red the actual value of pieces sold

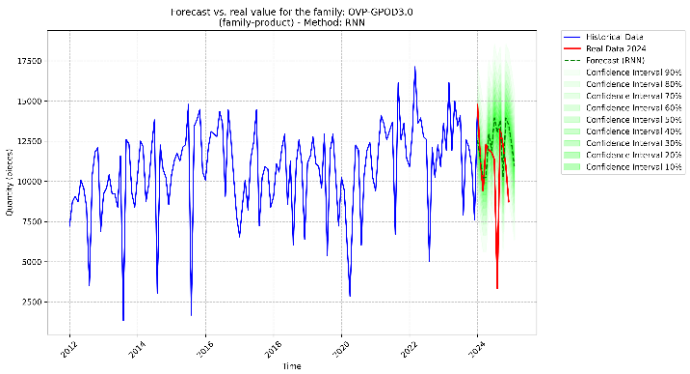


Figure 6.7: RNN model validation example - Family Product

The heatmap in the figure 6.8 shows the average percentage of correctly predicted products using the RNN model, across different forecast horizons (in months) and confidence levels.

Also in this case, Figure 6.9 shows that to obtain a reliable prediction in 75% of cases, a confidence band equal to 70% was necessary.

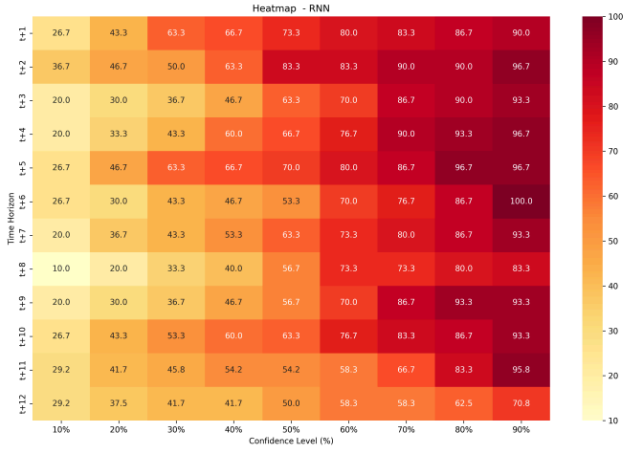


Figure 6.8: Heatmap of forecasting errors for the RNN model

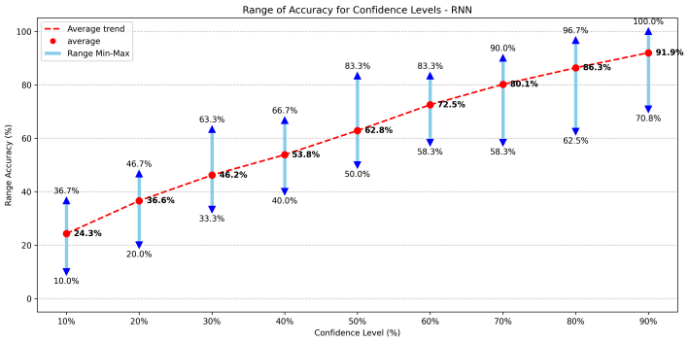


Figure 6.9: Level of accuracy of predictions for the RNN model

7 Conclusions

This study was designed to provide VHIT with an advanced and automated tool for forecasting the sales trends of its products. The objective was not merely to apply forecasting models to historical sales data but to develop an intelligent engine capable of classifying products based on their characteristics and dynamically adapting the forecasting methodology, taking into account possible changes over time.

To achieve this goal, we built a system that, using the updated sales dataset, can recognize the nature of the time series associated with each product and automatically select the most suitable forecasting model. The approach combines the best methodologies from scientific literature, offering a practical application of theoretical insights in an industrial setting.

The project was developed from its earliest stages through a close and continuous collaboration with VHIT, which played an active and essential role throughout the entire process. From the initial design of the solution to its concrete implementation, we maintained an ongoing dialogue through monthly meetings, held both remotely and in person at the Offanengo (CR) headquarters. This regular exchange made it possible to clearly define the project milestones, align expectations, and guide methodological choices in a way that was consistent with the company's operational needs.

The value of this research lies in its usefulness for manufacturing companies, which can derive numerous benefits from sales forecasting. One of the most significant advantages is inventory management: by leveraging confidence intervals from forecasts, it is possible to estimate the minimum quantity of products that should be kept in stock, reducing the risk of shortages or overstocking [Gardner 1990].

Another important advantage concerns financial planning. Having a reliable estimate of future sales allows the company to plan its production budget more precisely, optimizing resource allocation and reducing waste.

Regarding possible future developments of this work, our goal is to make the forecasting system even more sophisticated and tailored to VHIT's needs. The next step will be to integrate exogenous variables that can enrich the forecasting process, including market trends related to specific geographical areas, as well as detailed information about customers and the unique characteristics of individual products.

This evolution will further refine the forecasts, transforming them from purely mathematical computations into more robust and context-aware predictions. The idea is to combine the reliability of quantitative

analysis with insights gained from qualitative assessment, ensuring that the results reflect not only historical sales dynamics but also external influences that may impact future trends.

Bibliography

E. Babiloni, M. Cardós, J. M. Albarracín, and M.E. Palmer. DEMAND CATEGORISATION, FORECASTING, AND INVENTORY CONTROL FOR INTERMITTENT DEMAND ITEMS. *The South African Journal of Industrial Engineering*, 21(2), January 2012. ISSN 2224-7890, 1012-277X. doi: 10.7166/21-2-54. URL http://sajie.journals.ac.za/pub/article/view/54.

J E. Boylan and A A. Syntetos. Forecasting for Inventory Management of Service Parts. In *Complex System Maintenance Handbook*, pages 479–506. Springer London, London, 2008. ISBN 978-1-84800-010-0 978-1-84800-011-7. doi: 10.1007/978-1-84800-011-7_20. URL http://link.springer.com/10.1007/978-1-84800-011-7_20. Series Title: Springer Series in Reliability Engineering.

J E Boylan, A A Syntetos, and G C Karakostas. Classification for forecasting and stock control: a case study. *Journal of the Operational Research Society*, 59(4):473–481, April 2008. ISSN 0160-5682, 1476-9360. doi: 10.1057/palgrave.jors.2602312. URL https://www.tandfonline.com/doi/full/10.1057/palgrave.jors.2602312.

J D Croston. Forecasting and Stock Control for Intermittent Demands. *Operational Research Quarterly*, 23(3), 1972.

D Dhoka. ABC Classification for Inventory Optimization. *IOSR-JBM*, 15 (1):38–41, 2013. ISSN 23197668, 2278487X. doi: 10.9790/487X-1513841. URL http://iosrjournals.org/iosr-jbm/papers/Vol15-issue1/F01513841.pdf?id=7380.

A H C Eaves and B G Kingsman. Forecasting for the ordering and stock-holding of spare parts. *Journal of the Operational Research Society*, 55(4):431–437, April 2004. ISSN 0160-5682, 1476-9360. doi: 10.1057/palgrave.jors.2601697. URL https://www.tandfonline.com/doi/full/10.1057/palgrave.jors.2601697.

B E Flores. Multiple Criteria ABC Analysis. *International Journal of Operations & Production Management*, 7(3), 1987.

E S. Gardner. Evaluating Forecast Performance in an Inventory Control System. *Management Science*, 36(4):490–499, April 1990. ISSN 0025-1909, 1526-5501. doi: 10.1287/mnsc.36.4.490. URL https://pubsonline.informs.org/doi/10.1287/mnsc.36.4.490.

R Hemeimat, Lina Al-Qatawneh, Mazen Arafeh, and Shadi Masoud. Forecasting Spare Parts Demand Using Statistical Analysis. *AJOR*, 06(02):113–120, 2016. ISSN 2160-8830, 2160-8849. doi: 10.4236/ajor.2016.62014. URL http://www.scirp.org/journal/doi.aspx?DOI=10.4236/ajor.2016.62014.

C R. Schultz. Forecasting and Inventory Control for Sporadic Demand under Periodic Review. *The Journal of the Operational Research Society*, 38(5):453–458, 1987. ISSN 0160-5682. doi: 10.2307/2582735.

URL https://www.jstor.org/stable/2582735. Publisher: Palgrave Macmillan Journals.

A A. Syntetos and J E. Boylan. The accuracy of intermittent demand estimates. *International Journal of Forecasting*, 21(2):303–314, April 2005. ISSN 01692070. doi: 10.1016/j.ijforecast.2004.10.001. URL https://linkinghub.elsevier.com/retrieve/pii/S0169207004000792.

A A Syntetos, J E Boylan, and J D Croston. On the categorization of demand patterns. *Journal of the Operational Research Society*, 56(5):495–503, May 2005. ISSN 0160-5682, 1476-9360. doi: 10.1057/palgrave.jors.2601841. URL https://www.tandfonline.com/doi/full/10.1057/palgrave.jors.2601841.

R H. Teunter, A A. Syntetos, and M. Zied Babai. Intermittent demand: Linking forecasting to inventory obsolescence. *European Journal of Operational Research*, 214(3):606–615, November 2011. ISSN 0377-2217. doi: 10.1016/j.ejor.2011.05.018. URL https://www.sciencedirect.com/science/article/pii/S0377221711004437.

T. M. Williams. Stock Control with Sporadic and Slow-Moving Demand. *Journal of the Operational Research Society*, 35(10):939–948, October 1984. ISSN 0160-5682, 1476-9360. doi: 10.1057/jors.1984.185. URL https://www.tandfonline.com/doi/full/10.1057/jors.1984.185.

8 Appendix

8.1 Code Explanation

8.1.1 Data Cleaning Process

The data cleaning process was executed through a series of sequential functions, each outputting an intermediate file used as input for the next step.

Excel Sheets Consolidation

Function: `unisci_fogli_excel()`
Input: `dati_iniziali/VHIT_Actuals+2023.xlsx`
Output: `elaborazione_dati_iniziali/step1_unisci_sheet.xlsx`
Description: Merges all Excel sheets into a single DataFrame, selecting columns A through R.

Unified Date Column Creation

Function: `crea_colonna_data()`
Input: `elaborazione_dati_iniziali/step1_unisci_sheet.xlsx`
Output: `elaborazione_dati_iniziali/step2_colonnaunica_date.xlsx`
Description: Creates a unified date column using invoice date or posting date, calculates average dates for missing values.

Monthly Aggregation

Function: `ordina_e_aggregazione()`
Input: `elaborazione_dati_iniziali/step2_colonnaunica_date.xlsx`
Output: `elaborazione_dati_iniziali/step3_raggruppa_per_mese.xlsx`
Description: Aggregates data by product and month, summing quantities and revenues.

Negative Values Correction

Function: `correggi_valori_negativi_e_zeri()`
Input: `elaborazione_dati_iniziali/step3_raggruppa_per_mese.xlsx`
Output: `elaborazione_dati_iniziali/step4_senza_valori_negativi_-senza0.xlsx`
Description: Distributes negative values to previous positive entries, removes rows with zero values.

Product Family Creation

Inputs:

- `elaborazione_dati_iniziali/step4_senza_valori_negativi_senza0.xlsx`
- `dati_iniziali/PN_VHIT.xlsx`

Output: `elaborazione_dati_iniziali/step5_pn_vhit_with_status_check.xlsx`
Description: Creates product families based on product code, name, and client identifier; excludes obsolete products.

Completing Missing Months

Function: `completa_mesi_mancanti()`
Input: `elaborazione_dati_iniziali/step5_pn_vhit_with_status_check.xlsx`
Outputs:

- `elaborazione_dati_iniziali/finaldataset-family-product.xlsx`
- `elaborazione_dati_iniziali/finaldataset-family-product-client.xlsx`

Description: Adds missing months with zero quantity to complete the time series for each product family, produces two datasets for different aggregation levels.

8.1.2 General Statistics

Pareto Analysis

To gain insight into the revenue distribution patterns across product families, a Pareto analysis was conducted on both aggregation levels.

Data Preparation for Pareto Analysis

Input: `elaborazione_dati_iniziali/step5_pn_vhit_with_status_check.xlsx`
Outputs:

- `pareto_family_product.xlsx`
- `pareto_family_product_client.xlsx`

Description: Revenue data aggregation by product families, calculating total TNS and quantities for each family, computing percentage contribution to total revenue, and sorting families by revenue contribution from highest to lowest.

Pareto Chart Generation

Outputs:

- `statistiche_grafici/pareto_family_product.png`
- `statistiche_grafici/pareto_family_product_client.png`

Description: Creation of Pareto charts displaying revenue bars with cumulative percentage lines, showing the 80/20 distribution pattern across product families.

The resulting Pareto analyses are presented in Figures 3.2 and 3.3, which illustrate the concentration of revenue among the product families and demonstrate the typical Pareto distribution where a small percentage of families generates the majority of total revenue.

8.1.3 Classification

The classification of product families was performed through the analysis of time series data, categorizing each family based on demand patterns.

Data Preparation

Functions: `prepare_time_series()`, `classification()`
Input:

- `elaborazione_dati_iniziali/finaldataset-family-product.xlsx`
- `elaborazione_dati_iniziali/finaldataset-family-product-client.xlsx`

Description: Prepares time series by formatting dates and calculating ADI (Average Demand Interval) and CV^2 (squared Coefficient of Variation).

Stationarity Test

Function: `is_stationary()`
Description: Tests time series stationarity using the Dickey-Fuller test and determines appropriate forecasting method.

Demand Pattern Classification

Function: `classifica_famiglie()`
Outputs:

- `classificazione&forecasting/classified_family_product.xlsx`
- `classificazione&forecasting/classified_family_product_client.xlsx`

Description: Categorizes each product family into four classes based on ADI and CV^2 values:

- Smooth: Low ADI (< 1.32) and low CV^2 (< 0.49)
- Intermittent: High ADI (≥ 1.32) and low CV^2 (< 0.49)
- Erratic: Low ADI (< 1.32) and high CV^2 (≥ 0.49)
- Lumpy: High ADI (≥ 1.32) and high CV^2 (≥ 0.49)

Following the classification process, scatter plots were generated to visualize the distribution of product families across the four demand pattern categories based on ADI and CV^2 values.

ADI-CV^2 Scatter Plots

Outputs:

- `statistiche_grafici/scatter_adi_cv2_classified_family_product.png`
- `statistiche_grafici/scatter_adi_cv2_classified_family_product_client.png`

Description: Generation of scatter plots displaying each family's position in the ADI-CV^2 space with classification boundaries at ADI=1.32 and CV^2=0.49, color-coded by demand pattern classification.

The classification distribution for the Family Product Client aggregation level is illustrated in Figure 4.2, with the corresponding statistical breakdown presented in Table 4.2.

Classification Summary Statistics

Function: `create_classification_summary()`
Outputs:

- `statistiche_grafici/classification_summary_family_product.xlsx`
- `statistiche_grafici/classification_summary_family_product_client.xlsx`

Description: Calculation of family count and revenue percentages for each classification category, providing quantitative analysis of the distribution across Smooth, Intermittent, Erratic, and Lumpy demand patterns.

Pareto Analysis by Classification

The integration of classification results with Pareto analysis was performed to understand the revenue contribution of each demand pattern category.

Classified Pareto Charts

Function: `plot_pareto_fixed_legend()`
Inputs:

- Pareto analysis results
- Classification data from `classified_family_product.xlsx`
- Classification data from `classified_family_product_client.xlsx`

Outputs:

- `statistiche_grafici/pareto_classified_family_product_fixed.png`

- `statistiche_grafici/pareto_classified_family_product_client_fixed.png`

Description: Creation of enhanced Pareto charts with color-coded bars representing different demand pattern classifications, showing both revenue distribution and cumulative percentage lines.

The classified Pareto analyses are presented in Figures 4.3 and 4.4, demonstrating how different demand patterns contribute to the overall revenue distribution.

Distribution Analysis by Classification

Function: `create_pie_charts()`
Outputs:

- `statistiche_grafici/pie_families_product.png`
- `statistiche_grafici/pie_families_client.png`
- `statistiche_grafici/pie_revenue_product.png`
- `statistiche_grafici/pie_revenue_client.png`

Description: Generation of pie charts showing the distribution of families and revenue across the four classification categories, providing visual representation of the proportion of each demand pattern type.

The distribution analysis results are visualized in Figures 4.5, 4.6, 4.7, and 4.8, illustrating both the count and revenue share of each classification category across different aggregation levels.

8.1.4 Forecasting Methods

Different forecasting methods were applied based on the demand pattern classification and time series characteristics.

TSB Method

Function: `metodo_TSB()`
Description: Implements the Teunter-Syntetos-Babai method for intermittent demand, using exponential smoothing with parameters α and β to estimate demand size and probability.

SARIMA Model

Description: Applied to stationary time series, using grid search to find optimal SARIMA parameters (p,d,q)(P,D,Q,s) by minimizing AIC.

RNN Model

Description: Implements a stacked Recurrent Neural Network for non-stationary series, testing multiple configurations with different units, epochs, batch sizes, and dropout rates.

Forecasting Implementation

Function: `save_forecasting_data()`
Inputs:

- Classification results from `classifica_famiglie()`
- Original time series data

Outputs:

- `classificazione&forecasting/forecast_results_family_product.xlsx`
- `classificazione&forecasting/forecast_results_family_product_client.xlsx`

Description: Generates forecasts for each product family using the appropriate method (TSB, SARIMA, or RNN) based on classification, includes confidence intervals from 10% to 90%, and handles error cases and anomalies.

Individual Family Forecast Charts

Following the forecasting process, comprehensive visualization was implemented to display the forecast results for each product family, showing historical data, predictions, and confidence intervals.

Forecast Chart Generation

Function: `plot_family_forecast()`
Inputs:

- `classificazione&forecasting/forecast_results_family_product.xlsx`
- `classificazione&forecasting/forecast_results_family_product_client.xlsx`
- `elaborazione_dati_iniziali/finaldataset-family-product.xlsx`
- `elaborazione_dati_iniziali/finaldataset-family-product-client.xlsx`
- Pareto analysis results for ordering

Outputs:

- `statistiche_grafici/grafici_family_product/`
- `statistiche_grafici/grafici_family_product_client/`

Description: Generation of individual forecast charts for each product family, displaying historical time series data, forecast predictions, and multiple confidence intervals (10% to 90%). Charts are ordered by Pareto ranking and include method identification (TSB, SARIMA, or RNN).

Model-Specific Examples

Representative examples of the three forecasting methods are illustrated in the following Figures:

- TSB method example: Figure 5.2 demonstrates the TSB forecasting approach for intermittent demand patterns
- SARIMA method example: Figure 5.3 shows the SARIMA model application for stationary time series
- RNN method example: Figure 5.4 presents the Recurrent Neural Network forecasting for complex non-stationary patterns

8.1.5 Forecasting Validation

Following the forecasting phase, a validation process was implemented to assess prediction accuracy by comparing forecasts with actual 2024 sales data.

Data Integration

Inputs:

- `validazione_previsioni/finaldataset-family-product.xlsx`
- `validazione_previsioni/finaldataset-family-product-client.xlsx`
- Previous forecasting results from `classificazione&forecasting/`

Validation Visualization

Function: `plot_family_forecast_with_real()`
Outputs:

- `validazione_previsioni/grafici_family_product/`
- `validazione_previsioni/grafici_family_product_client/`

Description: Generation of validation charts displaying historical data, forecast predictions, actual 2024 values, and confidence intervals for performance assessment.

Validation Examples

Representative validation examples demonstrate the performance of each forecasting method:

- TSB model validation: Figure 6.1 shows the TSB forecasting approach for intermittent demand patterns
- SARIMA model validation: Figure 6.4 demonstrates the SARIMA model application for stationary time series
- RNN model validation: Figure 6.7 presents the Recurrent Neural Network forecasting for complex patterns

These validation examples provide direct visual comparison between predicted and actual values, enabling assessment of forecasting model accuracy and reliability across different demand pattern categories.

8.1.6 Comprehensive Results

Following the individual validation process, comprehensive statistical analysis was conducted to assess the overall reliability and accuracy of each forecasting method across all product families.

Statistical Accuracy Assessment

Functions: `analyze_forecast_accuracy()`, `create_accuracy_plots()`, `create_confidence_summary()`
Inputs:

- Real data from `validazione_previsioni/`
- Forecast results from `classificazione&forecasting/`
- Multiple confidence levels (10% to 90%)

Outputs:

- `validazione_previsioni/grafici_finali_family_product/`
- `validazione_previsioni/grafici_finali_family_product_client/`
- `validazione_previsioni/dati_family_product/`
- `validazione_previsioni/dati_family_product_client/`

Description: Generation of aggregated performance metrics evaluating prediction accuracy across different time horizons and confidence intervals for each forecasting method.

TSB Model Performance

The TSB intermittent demand model effectiveness is shown through:

- Error distribution heatmap: Figure 6.2
- Accuracy level analysis: Figure 6.3

SARIMA Model Performance

The SARIMA statistical model reliability is demonstrated by:

- Error distribution heatmap: Figure 6.5
- Accuracy level analysis: Figure 6.6

RNN Model Performance

The Recurrent Neural Network model performance is illustrated through:

- Error distribution heatmap: Figure 6.8
- Accuracy level analysis: Figure 6.9

CHAPTER 2

Improving the Performance of Traditional Interest Rates Term Structure Models Using Nature-Inspired Optimization Algorithms

Abstract

The Nelson-Siegel, Svensson, and De Rezende-Ferreira-Ferreira models represent some of the most widely adopted parametric frameworks for modeling the term structure of risk-free interest rates. Nonetheless, their reliability may diminish under volatile market conditions. Given the central role of the term structure in determining the time value of money and in pricing financial instruments, this study seeks to enhance the statistical robustness of these models by integrating nature-inspired global optimization algorithms, specifically Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA). These techniques are employed to explore the search space and identify promising regions, which are then used to initialize local optimization routines, improving convergence and reducing the risk to fall in a local minima. Finally, to validate our results against a challenging benchmark, we compare the performance of the parametric models with those obtained using a non-parametric machine learning approach, namely Gaussian Process Regression.

The analysis was conducted on three distinct datasets to assess the robustness of the results: the first includes currencies from five developed economies, the second comprises nine currencies from Pacific countries, and the third consists of sovereign yield curves from five European nations. The findings confirm that integrating nature-inspired global optimization algorithms substantially enhances the fitting performance of traditional parametric models, making them competitive with machine learning approaches a comparison that was previously unimaginable in this context.

Keywords: Interest rates term structure, Genetic algorithm, Particle swarm optimization, Gaussian process regression, Simulated annealing

JEL code: G17, E43, C45, C53

Acknowledgements

I gratefully acknowledge the valuable comments and suggestions received during the presentation of this study at several international conferences, including:

IFABS 2025 Oxford Conference (Saïd Business School, University of Oxford, UK, April 15–17, 2025),

The 2025 Modern Risk Society (MRS) International Risk Conference (Suffolk University, Boston, Massachusetts, July 24–26, 2025),

Corporate Governance Risk Management in Financial Institutions 2024 (CGRM 2024) (University of Bari Aldo Moro, Bari, Italy, July 11–12, 2024),

and the International Risk Management Conference 2024 (IRMC 2024) (Bocconi University, Milan, Italy, June 24–25, 2024).

I am especially thankful to the conference participants and anonymous reviewers for their constructive feedback, which has significantly improved the quality of this manuscript.

1 Introduction

The study of the term structure of interest rates is of central importance across various sectors of the economy and finance, playing a crucial role in the valuation of financial instruments, risk management, and the implementation of monetary policy. The shape of the yield curve is fundamental for calculating discount factors, which are used in pricing bonds, derivatives, structured products, and more generally, any financial instrument involving a projection of future cash flows. Given the importance of this tool, numerous techniques have been proposed over time to estimate the yield curve effectively and robustly. Among these, a widely adopted approach in both academic and professional settings is based on nonlinear parametric models. In particular, the models proposed by [Nelson and Siegel 1987], [Svensson 1994] and [De Rezende and Ferreira 2013] are currently considered standard references for modeling the term structure.

These models are regarded as mathematically robust and, moreover, they are based on the expectations theory, which makes them reliable also from the perspective of economic interpretation, as underlined and reiterated by [Litterman and Scheinkman 1991].

However, despite their numerous advantages, these models also present some significant methodological challenges. In particular, difficulties arise during the choice of the starting points to be used as parameters for the models. This process is crucial, as only with optimal starting values can the curves be solved correctly. There is no standardized or universally accepted procedure in the literature for identifying these initial values. In practice, local optimization methods are frequently employed. While these methods can be effective under stable market conditions, it has been observed that in scenarios involving high volatility or discontinuities, they often fail becoming trapped in local minima and thereby preventing proper curve calibration. Therefore, the issue of parameter calibration is still considered an unresolved problem and remains an open area of research.

In light of these limitations, this study aims to contribute to the existing literature by addressing the problem of dependency on starting values through a hybrid approach inspired by what are known in operations research as "memetic techniques". These methods rely on the synergistic combination of a global optimizer, capable of broadly exploring the solution space, with a local optimizer that refines the solution near the optimum. In our case, local optimization is already handled by the Levenberg-Marquardt and Trust Region algorithms. The innovation introduced lies in complementing this with a set of nature-inspired global

optimization algorithms, including two AI-inspired methods—Genetic Algorithms (GA) and Particle Swarm Optimization (PSO)—as well as a probabilistic metaheuristic, Simulated Annealing (SA). These algorithms are particularly well-suited to complex, high-dimensional solution spaces with multiple local minima and are designed to overcome the structural limitations of deterministic optimization techniques, thereby increasing the likelihood of identifying optimal starting points. Finally, to validate our results against a challenging benchmark, we compare the performance of the parametric models with those obtained using a non-parametric machine learning approach, namely Gaussian Process Regression.

Our study first confirms the strong dependence on starting values, as highlighted by [Cairns and Pritchard 2001], since a large portion of curves fail to converge under traditional models. This issue becomes increasingly evident when moving from the simpler Nelson-Siegel specification, with only four parameters, to the more complex De Rezende-Ferreira specification, which includes eight. Moreover, the inadequacy of traditional solvers in identifying appropriate starting points is particularly pronounced during periods of high volatility in the bond market, such as the post-COVID interest rate hikes.

The most significant outcome of our analysis lies in the ability of global solvers to effectively map the domain of feasible solutions, identifying reliable optima that can then be refined by local optimization algorithms, thereby enabling successful curve fitting. The inclusion of global solvers proves decisive, transforming a scenario where only a small fraction of curves could be resolved into one where the vast majority are successfully fitted. This leads to a marked improvement in performance, particularly for the more complex models such as Svensson and De Rezende-Ferreira, which previously delivered the weakest results without the support of global optimization techniques.

The improvement in performance is so substantial that these models become capable of competing with machine learning methodologies—a comparison that was previously out of reach.

The remainder of the paper is organized as follows: Section 2 presents a literature review focused on the study of the yield curve; Section 3 details the methodology, including models, algorithms, and techniques used in this research; Section 4 discusses the application studies and results obtained across different underlyings; and Section 5 concludes this second part of the thesis.

2 Literature Review

Given the significant importance of studying the shape of the interest rates term structure for both central banks and private market participants, this line of research has long attracted considerable attention. Indeed, a vast body of scientific literature has emerged over the past decades, dedicated to the accurate representation and modeling of this structure.

An early advanced approach, initially proposed by [Fama and Bliss 1987], is the smoothed bootstrap, which recommended deriving zero-coupon rates from raw market data and then fitting them with a smooth and continuous curve. To implement this idea, a wide range of spline fitting techniques have been employed, including quadratic and cubic splines [McCulloch 1971], as well as the methods proposed by [Mastronikola 1991] and [Deacon and Derry 1997], exponential splines [Vasicek and Fong 1982], B-splines [Shea 1984]; [Steeley 1991], quartic maximum smoothness splines [Adams and Van Deventer 1994], and penalty function-based splines [Fisher et al. 1994]; [Waggoner 1998].

However, after years of widespread use, several intrinsic limitations of spline-based techniques have become increasingly apparent. From a technical standpoint, one of the major criticisms is that splines tend to overfit the market data, capturing not only the underlying trends but also noise such as measurement errors, liquidity issues, or bid-ask spreads. Another significant weakness lies in their sensitivity to the placement and number of knots [Dahlquist and Svensson 1996]: even minor changes in knot positions can cause large variations in the shape of the curve, raising concerns about robustness. Perhaps the most fundamental drawback, however, is the lack of economic interpretability, spline parameters do not have any direct economic meaning, making it difficult to relate the model to financial theory or macroeconomic principles [Annaert et al. 2012]. In addition, [Seber 2004] also noted that these techniques could create a "black-box" interpretative effect, which makes them less desirable in stable or turbulence-free market conditions.

Once it became clear that spline-based techniques presented intrinsic and hardly surmountable limitations, attention shifted toward alternative approaches—specifically, nonlinear parametric models. A major turning point was the work of [Nelson and Siegel 1987], who, drawing on the expectations theory of the term structure of interest rates, proposed a parsimonious method to describe the entire yield curve using a small set of parameters while still maintaining a good fit to observed data. The empirical reliability and theoretical appeal of

this framework encouraged further extensions by [Svensson 1994] and later by [De Rezende and Ferreira 2013], who introduced additional parameters to enhance flexibility and improve the model's capacity to fit term structures even under high-volatility conditions.

However, expanding the number of parameters is not without drawbacks. The main objective of parametric yield curve models is to provide a parsimonious representation of the term structure—capturing as much market information as possible with the fewest parameters. This principle inherently creates a trade-off between model simplicity and fitting accuracy. While adding parameters generally improves the fit, this is only justified if the gain is economically meaningful. For this purpose, model selection criteria such as the Schwarz-Bayes Criterion [Wei 1990]; [Cairns and Pritchard 2001] are commonly employed to evaluate the net benefit of including additional factors.

[Nelson and Siegel 1987], followed by [Svensson 1994] and later [De Rezende and Ferreira 2013], explored the possibility of achieving a smooth bootstrap through nonlinear regression models, capable of fitting a wide range of yield curve configurations. Beyond interest rates term structures, these models have also been successfully applied to credit default swap premiums and commodity prices [Chu et al. 2022], demonstrating remarkable versatility. Their empirical robustness and economic interpretability—highlighted by [Litterman and Scheinkman 1991] have secured their adoption in both academia and practice. Indeed, the Federal Reserve has employed the Svensson model for constructing daily U.S. Treasury yield curves since 1980, using the Nelson-Siegel specification for earlier periods with fewer securities available [Gürkaynak et al. 2007]. Similarly, the European Central Bank has relied on both Nelson-Siegel and Svensson extensions to estimate euro area government bond yield curves [Brousseau 2002].

The use of nonlinear parametric models has, however, highlighted another critical issue, this time not related to the shape of the generated curve, but to the parameter calibration process, which has proven to be anything but straightforward. Even in the [Nelson and Siegel 1987] model, estimating the coefficients requires particular attention; in later specifications—including those of [Svensson 1994] and [De Rezende and Ferreira 2013], which introduce a greater number of parameters—the problem has become even more pronounced. In fact, these models display an increasing sensitivity to the choice of starting points. To address this difficulty, numerical optimization routines have been widely adopted, often based on local search algorithms such as the quasi-Newtonian L-BFGS method [Nocedal 1980], or on direct search methods such as the [Nelder and Mead 1965]. Hybrid procedures, such as those proposed by [Manousopoulos and Michalopoulos 2009], have also been investigated in order to limit such disadvantages. However, even when optimization techniques are employed, the problem is not entirely solved. [Cairns and Pritchard 2001] demonstrate that one cannot start from just any set of points—appropriate initialization remains essential. Furthermore, a significant instability in the estimated time series of coefficients has been documented, often accompanied

by excessively large standard errors, which undermines the empirical reliability of the models.

In response to these challenges, attempts have been made to adopt alternative methodologies that, in recent years, have become increasingly popular in the financial sector. These rely on more innovative approaches, such as machine learning and artificial intelligence techniques [Ahmed et al. 2022]. Several studies in the literature have pursued a bottom-up approach, using ML paradigms to analyze data without making a priori assumptions about the functional form of the interest rate curve. For example, the radial basis function neural network [Cafferata et al. 2019], the feed-forward neural network [Giribone and Caligaris 2015], and kernel ridge regression [Filipovic et al. 2022] have been examined to evaluate their effectiveness. The results show that these techniques are capable of fitting the curve with lower error compared to traditional parametric models.

Nevertheless, a degree of skepticism remains, since machine learning approaches are not based on any predefined functional form and therefore lack an intrinsic economic interpretation; their performance is entirely dependent on the dataset on which they are trained. Although the reference dataset may be rich and informative, it does not guarantee interpretability. For this reason, the use of machine learning techniques—particularly those related to deep learning—must always be assessed in light of market needs and data availability [Cafferata et al. 2018].

Other recent applications of ML in this domain can be found in [Bie et al. 2025], where a tree-based technique is applied to identify macroeconomic regimes using observable variables such as the unemployment rate and the yield on 3-month Treasury bills, with the aim of generating forecasts for the interest rate curve estimated using the Nelson-Siegel model.

3 Methodology

3.1 Financial Bootstrapping

This study analyzes the interest rate curve across three distinct contexts. In one of them, the curves refer to sovereign states (France, Germany, Italy, Spain, United Kingdom), for which we rely on government bonds issued by central banks with a wide range of maturities. In this case, there is no need to apply a bootstrapping process, as yields for different maturities are directly available and observable in the market.

In the other two contexts, we focus on fitting the interest rate curve for various currencies. The currencies analyzed are grouped as follows:

- Developed currencies: CHF (Swiss Franc), EUR (Euro), GBP (British Pound), JPY (Japanese Yen), USD (US Dollar)
- Asia-Pacific and emerging markets: AUD (Australian Dollar), CNY (Chinese Yuan), HKD (Hong Kong Dollar), INR (Indian Rupee), JPY (Japanese Yen), MYR (Malaysian Ringgit), NZD (New Zealand Dollar), and TWD (New Taiwan Dollar), in addition to USD (US Dollar)

In these cases, we construct the curve based on market-traded financial instruments such as Deposits, Futures, Forward Rate Agreements (FRA) and Swaps. This type of risk-free curve offers several advantages: it tends to reflect the actual market sentiment more closely, reacts more quickly to changing economic conditions, and is generally less affected by macroeconomic or political distortions that may influence government bond yields.

In such cases, the yields associated with different maturities are not directly observable and must be inferred implicitly through a financial bootstrapping process. This technique allows us to derive the implied interest rate for each maturity based on the prices of traded instruments, as described by [Hagan and West 2006].

To reconstruct the interest rate curves associated with the various currencies considered in the different case studies, we retrieved market-listed instrument prices from Bloomberg®. The data used reflect the actual instruments available for each currency; for a complete overview of which instruments were employed in each curve, please refer to Tables 4.1 and Figure 4.5

The reference instruments used for curve construction are: deposits, futures, and swaps. Each category covers specific maturity segments and provides relevant information for different parts of the curve.

Deposits

Interbank deposits are typically available for very short maturities, ranging from a few days to several months. These instruments allow us to directly compute discount factors and short-term spot rates using the following formulas:

$$DF(t) = \frac{1}{1 + r(t) \cdot \Delta t} \tag{3.1}$$

$$R(t) = \frac{1}{\Delta t} \cdot \ln\left(\frac{1}{DF(t)}\right) \tag{3.2}$$

where:

- $DF(t)$ is the discount factor at time t;
- $r(t)$ is the interest rate;
- Δt is the time to maturity expressed in years;
- $R(t)$ is the continuously compounded spot rate at time t.

Futures

Interest rate futures, such as Euribor or Fed Funds futures, are used to estimate implied forward rates over intermediate horizons. Unlike deposits, they provide information about expected future returns over specific periods. The corresponding discount factors and implied rates can be derived as follows:

$$DF(t_1, t_2) = \frac{1}{1 + f_{t_1,t_2} \cdot \Delta t} \tag{3.3}$$

$$R(t_0, t_2) = \frac{1}{t_2 - t_0} \cdot \ln\left(\frac{1}{DF(t_0, t_2)}\right) \tag{3.4}$$

where:

- $DF(t_1, t_2)$ is the discount factor between time t_1 and t_2;
- f_{t_1,t_2} is the forward rate applicable over the interval $[t_1, t_2]$;
- Δt is the time difference $t_2 - t_1$, expressed in years;
- $R(t_0, t_2)$ is the continuously compounded spot rate from time t_0 to t_2.

Swaps

Interest rate swaps are derivative instruments that typically cover medium- to long-term maturities, ranging from 2 year up to 30 years or more. However, in several cases, they are also available for intermediate maturities, overlapping partially with futures. The bootstrapping process allows us to extract the remaining discount factors from the observed market swap rates using the following formulas:

$$DF_n = \frac{1 - \sum_{i=1}^{n-1} \Delta_i \cdot DF_i}{\Delta_n} \tag{3.5}$$

$$R_n = \frac{1}{T_n} \cdot \ln\left(\frac{1}{DF_n}\right) \tag{3.6}$$

where:

- DF_n is the discount factor for the n-th maturity;
- Δ_i is the year fraction (accrual period) associated with the i-th cash flow;
- DF_i is the discount factor for the i-th maturity, already computed;
- Δ_n is the year fraction for the n-th cash flow;
- R_n is the continuously compounded spot rate for maturity T_n;
- T_n is the time to maturity (in years) corresponding to the n-th period.

3.2 Nonlinear Parametric Models

Once the yields (i.e. the zero or spot rates) across different maturities have been derived through the financial bootstrapping technique, we proceed to estimate the yield curves for the selected currencies. Among the various methodologies available for yield curve modeling, we adopt a class of nonlinear parametric models, including those proposed by [Nelson and Siegel 1987], [Svensson 1994], and [De Rezende and Ferreira 2013]. The choice of these specifications is motivated by both their solid economic interpretability and their widespread use: they currently represent the standard approach adopted by central banks, financial institutions, and international monetary authorities for modeling the term structure of interest rates.

3.2.1 Nelson-Siegel model (1987)

[Nelson and Siegel 1987] were among the first to propose a simple yet effective parametric model for representing the entire term structure of interest rates. The model is based on the empirical observation that yield curves tend to exhibit recurring shapes—monotonic, humped,

or S-shaped—and aims to replicate them through a flexible but parsimonious functional form. The authors rely on functions derived from differential equation solutions, consistent with the expectations hypothesis, which treats forward rates as predictors of future spot rates. The Nelson-Siegel model expresses the yield $R(t)$ at maturity t as:

$$R(t) = \beta_0 + \beta_1 \cdot \frac{\tau\left[1 - \exp\left(-\frac{t}{\tau}\right)\right]}{t} + \beta_2 \cdot \left[\frac{\tau\left[1 - \exp\left(-\frac{t}{\tau}\right)\right]}{t} - \exp\left(-\frac{t}{\tau}\right)\right] \tag{3.7}$$

where:

- β_0 defines the long-term level of the yield curve (asymptote), typically $\beta_0 > 0$;
- β_1 captures the short-term component (decays quickly to zero);
- β_2 models the medium-term component (hump or curvature);
- $\tau > 0$ is a time-decay parameter that determines the location of the hump.

3.2.2 Svensson model (1994)

Building on the Nelson-Siegel framework, [Svensson 1994] proposes an extended parametric model aimed at improving the flexibility and accuracy of yield curve fitting. His motivation arises from the need to estimate forward rates from bond price data in the absence of explicit forward markets—a task previously addressed by [McCulloch 1971] using cubic spline techniques. However, Svensson highlights a key drawback of McCulloch's approach: the instability of forward rate estimates, particularly at long maturities.

To overcome this issue, Svensson adopts the Nelson-Siegel functional form and introduces a fourth term, enhancing the model's ability to capture more complex yield curve shapes. This additional term allows for greater adaptability in fitting structures with multiple humps or kinks, where the standard Nelson-Siegel model may fall short. The resulting model balances parsimony with flexibility, making it suitable for a wider range of empirical yield curve configurations.

$$R(t) = \beta_0 + \beta_1 \cdot \frac{\tau_1\left[1 - \exp\left(-\frac{t}{\tau_1}\right)\right]}{t} + \beta_2 \cdot \left[\frac{\tau_1\left[1 - \exp\left(-\frac{t}{\tau_1}\right)\right]}{t} - \exp\left(-\frac{t}{\tau_1}\right)\right] + \beta_3 \cdot \left[\frac{\tau_2\left[1 - \exp\left(-\frac{t}{\tau_2}\right)\right]}{t} - \exp\left(-\frac{t}{\tau_2}\right)\right] \tag{3.8}$$

where:

- β_3 determines the dimension and direction of the secondary hump.
- τ_2 specifies the position of the secondary hump in time.

3.2.3 De Rezende-Ferreira-Ferreira model (2011)

Extending the Nelson-Siegel-Svensson framework, [De Rezende and Ferreira 2013] propose a five-factor model designed to further increase the flexibility of yield curve fitting. Their formulation builds directly on the [Svensson 1994] extension by adding a fifth term—interpreted as a second slope component—to capture more intricate term structure dynamics. This additional factor enhances the model's capacity to accommodate highly non-linear and twisted yield curves, which are often observed in emerging markets. The five-factor model addresses the limitations of earlier specifications proposed by [Nelson and Siegel 1987] and [Svensson 1994], offering improved in-sample fit and predictive power—particularly in contexts where standard parametric forms may fail to capture local inflection points in the yield curve.

$$
\begin{aligned}
R(t) = \beta_0 + \beta_1 \cdot & \frac{\tau_1 \left[1 - \exp\left(-\frac{t}{\tau_1}\right)\right]}{t} \\
& + \beta_2 \cdot \left[\frac{\tau_1 \left[1 - \exp\left(-\frac{t}{\tau_1}\right)\right]}{t} - \exp\left(-\frac{t}{\tau_1}\right)\right] \\
& + \beta_3 \cdot \left[\frac{\tau_2 \left[1 - \exp\left(-\frac{t}{\tau_2}\right)\right]}{t} - \exp\left(-\frac{t}{\tau_2}\right)\right] \\
& + \beta_4 \cdot \left[\frac{\tau_3 \left[1 - \exp\left(-\frac{t}{\tau_3}\right)\right]}{t} - \exp\left(-\frac{t}{\tau_3}\right)\right]
\end{aligned} \tag{3.9}
$$

where:

- β_4 determines the dimension and direction of the tertiary hump.
- τ_3 specifies the position of the tertiary hump in time.

3.3 Optimizers

From the outset, scholars have recognized that the calibration of yield curve models is highly sensitive to the choice of initial parameter values. The problem of selecting appropriate starting points for nonlinear optimization has been a central concern, given its influence on the convergence and stability of the estimation process. [Nelson and Siegel 1987] addressed this challenge by employing a grid search over the

decay parameter τ, solving for the remaining coefficients via non linear least squares at each grid point. [Svensson 1994] proposed minimizing the sum of squared errors using methods such as nonlinear least squares or maximum likelihood, both of which require suitable initial guesses for successful optimization. [De Rezende and Ferreira 2013] adopted a similar approach, conducting a discrete search over candidate decay parameters to minimize the root mean squared error (RMSE), thus avoiding arbitrary parameter fixing.

Despite these efforts, there is no universally accepted strategy for solving the underlying nonlinear optimization problem. In this study, we implement two widely used algorithms—Levenberg–Marquardt and Trust Region—which are considered standard optimization methods in MATLAB and are well-suited for calibrating parametric yield curve models.

3.3.1 Levenberg-Marquardt algorithm

The problem of nonlinear least squares optimization arises frequently in a wide range of applications, particularly in nonlinear curve fitting. The main difficulty lies in the fact that the model is nonlinear with respect to its parameters, which prevents the derivation of a closed-form solution. Therefore, iterative numerical methods must be used. Among these, gradient descent is simple and stable but often slow, while the Gauss–Newton method tends to be faster but can fail if the model is too complex or the starting point is too far from the optimum.

[Levenberg 1944] proposed a novel approach to improve convergence in this context. His idea was to interpolate between gradient descent and the Gauss–Newton method, aiming to combine the robustness of the former with the efficiency of the latter. This led to the development of an algorithm that updates the parameter vector $\mathbf{x}_k$ at iteration k according to the following rule:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (J^\top J + \lambda I)^{-1} J^\top F(\mathbf{x}_k) \tag{3.10}$$

In this formulation:

- J is the Jacobian matrix of the vector-valued function $F(\mathbf{x})$, containing the partial derivatives of the residuals with respect to the parameters;
- $F(\mathbf{x}_k)$ is the residual vector at iteration k;
- λ is the damping parameter that regulates the step size;
- I is the identity matrix.

The behavior of the algorithm is strongly influenced by the value of λ: when $\lambda \to 0$, the algorithm behaves like the Gauss–Newton method, favoring faster but potentially unstable updates. On the other hand, when $\lambda \to \infty$, it approximates gradient descent, which is more stable but slower. The algorithm dynamically adjusts λ at each iteration: it increases the damping if the error worsens (making the step more

conservative), and decreases it when the error improves (allowing faster convergence). This makes the method robust and capable of converging even when starting far from the optimum.

[Marquardt 1963] improved the method by addressing a limitation in Levenberg's original formulation. He observed that the isotropic penalization introduced by the identity matrix I was not sufficient to efficiently navigate narrow valleys in the error surface, known as "error valleys", where convergence may become unstable or excessively slow. Marquardt proposed replacing I with the diagonal of the Gauss–Newton approximation of the Hessian matrix, $J^\top J$, thereby incorporating local curvature information into the damping term. The updated iteration rule becomes:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - (J^\top J + \lambda \cdot \mathrm{diag}(J^\top J))^{-1} J^\top F(\mathbf{x}_k) \tag{3.11}$$

This modification allows the algorithm to scale the update step directionally: components corresponding to steeper curvatures are penalized more heavily, while flatter directions are less constrained. As a result, the algorithm retains the stability of Levenberg's original method while enhancing convergence performance in ill-conditioned problems.

The algorithm proceeds iteratively. Starting from an initial guess $\mathbf{x}_0$, the residual vector $\mathbf{r}_0$ is computed, and the parameters are updated using Marquardt's formulation. If the new residual $\mathbf{r}_{k+1}$ increases, the step is rejected, λ is increased, and the update is retried. If the residual decreases, the update is accepted and λ is reduced. This iterative adjustment continues until convergence is reached, typically when the change in the residual becomes sufficiently small.

In summary, the Levenberg–Marquardt algorithm provides an effective compromise between stability and convergence speed. By adapting to the local geometry of the objective function, it is well-suited for fitting nonlinear parametric models, even in the presence of complex data structures or noisy observations. Although it does not guarantee the identification of a global minimum, it reliably converges to a local minimum under broad conditions, which has made it a widely adopted tool in both academic and industrial applications.

3.3.2 Trust Region algorithm

The trust region method is an optimization strategy for unconstrained problems that relies on approximating the objective function with a simpler, more reliable model within a limited portion of the variable space. Unlike line-search methods, which fix a descent direction and then determine an optimal step length along that direction, the trust region method first defines a local neighborhood (the *trust region*) where the model is assumed to be an adequate approximation of the true function.

Given the current point x_k, a model $m_k(p)$ is constructed, typically a quadratic function obtained through a second-order Taylor expansion of $f(x)$ around x_k:

$$m_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T B_k p \tag{3.12}$$

where:

- $\nabla f(x_k)$ is the gradient of f at x_k,
- B_k is a symmetric matrix approximating the Hessian $\nabla^2 f(x_k)$,
- p is the displacement vector from x_k.

The trust region is defined as the set of displacements p satisfying $\|p\| \leq \Delta_k$, where $\Delta_k > 0$ is the current trust-region radius. At each iteration, the subproblem to be solved is:

$$\min_{p \in \mathbb{R}^n} m_k(p) \quad \text{subject to} \quad \|p\| \leq \Delta_k \tag{3.13}$$

The solution p_k of this subproblem provides a candidate update $x_{k+1} = x_k + p_k$. The quality of this step is evaluated by comparing the actual decrease in the function with the decrease predicted by the model:

$$\rho_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)} \tag{3.14}$$

The value of ρ_k governs the adaptation of the trust-region radius:

- If ρ_k is sufficiently large (e.g., $\rho_k > 0.75$), the model is considered reliable, and Δ_k is increased.
- If ρ_k is small (e.g., $\rho_k < 0.25$), the model fails to predict the function behavior, and Δ_k is reduced.
- If ρ_k is in between, Δ_k is kept unchanged.

The step p_k is accepted only if it provides a genuine decrease in the objective function; otherwise, it is rejected, and a new step is computed with a smaller trust region.

This approach dynamically adapts to the local geometry of the function, offering more robustness than linear methods, especially when the objective function is ill-conditioned or highly nonlinear. Although each iteration is computationally more expensive, the method is highly effective in achieving precise minimization.

3.4 Memetic Technique

The estimation through nonlinear parametric models presents a central challenge: the sensitivity of the calibration process to initial parameter values. Even when aided by local optimization routines such as Levenberg–Marquardt or Trust Region methods, these models often suffer from convergence to suboptimal local minima or exhibit instability across different initializations. This issue has been documented, for instance, by [Cairns and Pritchard 2001] in their empirical investigations.

To mitigate these limitations and enhance the robustness of the fitting procedure, we adopt a well-established strategy in the field of operations research and global optimization: the use of memetic techniques. This class of hybrid methods integrates global search mechanisms with local optimization, resulting in a cooperative framework that combines the exploratory power of evolutionary strategies with the refinement ability of deterministic optimizers. In our case, the global search will be performed by two Nature-Inspired algorithms—Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)—which will be coupled with the Trust Region method as a local optimizer.

Memetic techniques—named after the concept of memes introduced by [Dawkins 1976] operate on the principle of division of labor. Global algorithms are effective in exploring the solution space and in escaping local traps, but typically suffer from slow convergence once they reach promising regions. Local algorithms, conversely, exhibit fast convergence near local optima, but lack mechanisms to escape from poor starting points. A memetic approach resolves this trade-off by generating a population of candidate solutions through global exploration, each of which is subsequently refined via a local optimization step. This hybridization balances exploration and exploitation, yielding a flexible yet robust optimization scheme, as recognized since the early work of [Moscato 1989].

3.5 Nature-Inspired algorithms

Among stochastic optimization techniques, a class of algorithms of increasing importance is represented by Nature-Inspired Probabilistic Metaheuristics. These computational approaches derive their logic and operators from complex processes observed in nature. The fundamental intuition is that nature, through millions of years of "experimentation", has refined robust mechanisms for solving complex optimization and adaptation problems in uncertain and dynamic environments. By transposing these principles into an algorithmic context, analogous computational challenges can be addressed.

Within this broad paradigm of nature-inspired probabilistic metaheuristics, an important subcategory is that of Evolutionary Algorithms (EAs). These algorithms artificially simulate the fundamental mechanisms of Darwinian evolution, modeling concepts such as natural selection, genetic variation (through mutation and recombination oper-

ators), and inheritance. The idea is to evolve a "population" of candidate solutions, generation after generation, towards increasingly promising configurations in the solution space. Well-known examples of evolutionary algorithms include Genetic Algorithms (GA) and Particle Swarm Optimization (PSO).

Alongside evolutionary algorithms, there exist other nature-inspired metaheuristics that are not based on biological evolutionary principles or swarm behavior, but rather on other natural dynamics. A prominent example is Simulated Annealing (SA), which draws inspiration from the physical process of material annealing (tempering).

3.5.1 Genetic Algorithms

Within the family of evolutionary algorithms, Genetic Algorithms (GAs) represent one of the most successful techniques. They were primarily developed by [Holland 1975] and [De Jong 1975], drawing direct inspiration from the principles of natural selection and Darwinian genetics.

The core idea behind GAs is to simulate the evolution of a population of candidate "individuals" to search for an optimal solution to a given problem. Each individual represents a potential solution and is encoded as a "string" of genes.

The functioning of a GA proceeds through an iterative cycle that mimics the evolutionary process:

- **Population Initialization:** An initial population of individuals is generated randomly.
- **Fitness Evaluation:** Each individual in the population is evaluated using a fitness function.
- **Selection:** Based on fitness values, parent individuals are selected with higher probability of reproducing.
- **Genetic Operators (Crossover and Mutation):**
  - **Crossover (Recombination):** Mimics reproductive mechanisms by combining genetic material from two parent individuals to generate new offspring with hybrid characteristics.
  - **Mutation:** Introduces small random changes in the offspring's genetic strings.
- **Creation of the New Population:** The new individuals generated through crossover and mutation form the next generation.
- **Termination:** The cycle repeats for a predefined number of generations, or until a satisfactory solution or convergence criterion is met.

3.5.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) models the social and collective behavior of a swarm—such as a flock of birds or a school of fish—for optimization purposes. Introduced by [Kennedy and Eberhart 1995], and later refined by [Shi and Eberhart 1998], PSO operates on a population of particles that iteratively move through the problem's search space.

The iterative process of PSO involves evaluating the objective function at each step n for every particle at its current position. Based on these evaluations and shared information within the swarm, each particle's velocity is updated, thus determining its new position. This cycle repeats, enabling the population to converge toward promising regions of the search space.

The movement dynamics of each particle are influenced by three key components:

- **Best personal position** $s(n)$: The position in the search space that the individual particle has found so far, yielding the best objective function value.
- **Best global or neighborhood position** $g(n)$: The best position found by any particle in the entire swarm or within its specific neighborhood.
- **Previous velocity** $v(n-1)$: The inertia of the particle's previous movement.

The position of a particle $x(n)$ is updated at each iteration according to its velocity:

$$x(n+1) = x(n) + v(n) \tag{3.15}$$

This position is then adjusted, if necessary, to remain within the bounds of the search space.

The velocity of a particle $v(n+1)$ is updated approximately according to the following equation:

$$v(n+1) = w(n)v(n) + c_1 r_1(s(n) - x(n)) + c_2 r_2(g(n) - x(n)) \tag{3.16}$$

Where:

- $w(n)$ is the inertia weight, which balances the influence of the particle's previous velocity and may vary during the iterations to promote exploration (high values) or exploitation (low values).
- c_1 and c_2 are acceleration coefficients (or "cognitive and social weights"), positive constants that determine the influence of the particle's own experience and that of the swarm, respectively.
- r_1 and r_2 are independent random scalar values between 0 and 1, introducing stochasticity into the particle's learning process and allowing for more diverse exploration.

The combination of these terms enables each particle to balance its tendency to explore new areas of the search space with its attraction to the best solutions found both individually and collectively.

3.5.3 Simulated Annealing

Simulated Annealing (SA), introduced by [Kirkpatrick et al. 1983], emulates the physical process of annealing in metallurgy, where a slow cooling from high temperatures allows a system to reach a minimum energy state, reducing defects. Similarly, the SA algorithm aims to converge toward the global minimum of an objective function.

The working principle of Simulated Annealing is driven by an iterative process governed by a key parameter: the temperature T, which decreases progressively according to a predefined annealing schedule. At each iteration, the process unfolds as follows:

- **Generation of Trial Point:** A new trial point is randomly generated . The distance from the current position and the extent of the search are determined by a probability distribution whose scale is proportional to the current temperature T.
- **Acceptance Criterion:** The trial point is accepted as the next position if it improves the objective function. If the new point worsens the objective function, it may still be accepted with a probability given by an acceptance function. This probabilistic mechanism is essential to escape local minima and explore the search space globally. The acceptance probability P is given by:

$$P(\text{acceptance}) = \frac{1}{1 + \exp\left(\frac{\Delta}{T}\right)} \tag{3.17}$$

  where:

$$\Delta = f(\text{new point}) - f(\text{current point})$$

  When $\Delta > 0$ (i.e., the new point is worse), the acceptance probability lies between 0 and 0.5. Lower temperature T or larger Δ result in lower acceptance probabilities.

- **Temperature Update:** The system's temperature is systematically reduced based on a predefined schedule. Let k be the annealing index (proxy for the number of iterations). Common schedules include:
  - **Exponential decay:** $T = T_0 \cdot \alpha^k$
  - **Fast decay:** $T = \frac{T_0}{k}$
  - **Boltzmann decay:** $T = \frac{T_0}{\log(k)}$
- **Reannealing:** To prevent trapping in local minima, the algorithm may implement periodic reannealing by increasing the temperature after accepting a certain number of new points. This reactivates exploration. The annealing index k_i for component i is

adjusted to increase the temperature along that dimension, often based on estimated gradients of the objective function. A typical update is:

$$k_i = \log\left(\frac{T_i}{T_{0i}}\right) \cdot \frac{s_i}{\max_j(s_j)} \tag{3.18}$$

where:

- k_i is the annealing index for component i
- T_{0i} is the initial temperature of component i
- T_i is the current temperature of component i
- s_i is the product of the estimated gradient in direction i and the range in that direction

The algorithm continues to iterate while tracking the best solution found. The process stops when the average change in the objective function falls below a predefined tolerance threshold.

3.6 Gaussian Process approach

After attempting to improve the fitting performance of nonlinear parametric models through the use of metaheuristic techniques combining global and local optimizers in synergy, we now aim to compare the results obtained with those achievable using a completely different approach based on machine learning techniques.

Although these methods may fall short in terms of the economic interpretability of the curve, they are mathematically robust. As shown by [Delucchi and Giribone 2023], techniques such as Gaussian Process Regression (GPR) achieve excellent fitting performance, making them a challenging benchmark for comparison.

The working principle of Gaussian Process Regression (GPR) can initially be understood through the perspective of the *weight space view* [Rasmussen and Williams 2008], and then naturally evolves toward the more flexible *function space view* through the use of kernel functions. A GPR model is inherently nonparametric and probabilistic, meaning it does not assume a predefined functional form. This characteristic makes it significantly more flexible and adaptable than parametric models, enabling it to effectively capture highly nonlinear relationships.

Let us consider a training dataset

$$D = \{(x_i, y_i)\}_{i=1}^{n} \tag{3.19}$$

where $x_i \in \mathbb{R}^d$ represents the input vector and $y_i \in \mathbb{R}$ is the corresponding observed target value.

The basic linear regression model is formulated as:

$$y = x^\top \beta + \epsilon \tag{3.20}$$

where:

- x: input vector.
- β: vector of model coefficients (weights).
- ϵ: additive noise, assumed to follow a Gaussian distribution with zero mean and variance σ^2, i.e., $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

This noise assumption, combined with the model structure, defines the likelihood of the observations given the parameter configuration:

$$p(y \mid X, \beta, \sigma^2) = \mathcal{N}(y \mid X\beta, \sigma^2 I) \tag{3.21}$$

where:

- y: vector of observed target values.
- X: matrix of training inputs.
- I: identity matrix.

In the Bayesian formalism, a prior distribution is introduced on the weights $\boldsymbol{\beta}$, expressing initial beliefs about the parameters before observing the data. Typically, a zero-mean Gaussian prior is assumed:

$$p(\beta) = \mathcal{N}(\beta \mid 0, \Sigma_p),$$

where Σ_p is the prior covariance matrix.

Applying Bayes' theorem allows us to compute the posterior distribution over the parameters β, which integrates the information from the likelihood and the prior, summarizing all knowledge about the weights after observing the data. Given Gaussian priors and likelihoods, the posterior is also Gaussian, with analytically derivable mean μ_{post} and covariance Σ_{post}.

3.6.1 Extension via Basis Functions and Latent Gaussian Processes

To overcome the limited expressivity of linear models and handle nonlinear relationships, the inputs x are projected into a high-dimensional feature space through a set of basis functions $h(x)$. The GPR model introduces latent variables $f(x_i)$ derived from a Gaussian Process (GP), in combination with these explicit basis functions.

A Gaussian Process $\{f(x) \mid x \in \mathbb{R}^d\}$ is a collection of random variables, any finite subset of which follows a joint Gaussian distribution. A GP is fully specified by its mean function $m(x)$ and covariance function (or kernel) $k(x, x')$, where:

$$\mathbb{E}[f(x)] = m(x), \quad \text{Cov}[f(x), f(x')] = k(x, x') \tag{3.22}$$

The GPR model is formulated as:

$$y = h(x)^\top \beta + f(x) + \epsilon \tag{3.23}$$

where:

- $h(x)$: vector of basis functions mapping x into a feature vector in $\mathbb{R}^p$.
- β: $p \times 1$ vector of coefficients for the basis functions.
- $f(x) \sim \mathcal{GP}(0, k(x, x'))$: zero-mean Gaussian Process with covariance function $k(x, x')$, capturing the smoothness and non-parametric nature of the response.
- ϵ: observation noise.

Thus, the probabilistic model for each response instance y_i is:

$$p(y_i \mid f(x_i), x_i) \sim \mathcal{N}(y_i \mid h(x_i)^\top \beta + f(x_i), \sigma^2) \tag{3.24}$$

The vectorized form of the model for the entire training set is:

$$p(y \mid f, X) \sim \mathcal{N}(y \mid H\beta + f, \sigma^2 I) \tag{3.25}$$

where:

- X: matrix of training inputs.
- y: vector of observed target values.
- H: matrix of basis functions evaluated at training inputs.
- f: vector of latent variables corresponding to the training observations.

The joint distribution of the latent variables f in the GPR model is:

$$p(f \mid X) \sim \mathcal{N}(f \mid 0, K(X, X)) \tag{3.26}$$

where $K(X, X)$ is the covariance matrix (or kernel matrix) with elements $K_{ij} = k(x_i, x_j)$. The covariance function $k(x, x')$ is typically parameterized by a set of hyperparameters θ and is often explicitly written as $k(x, x' \mid \theta)$.

3.6.2 Parameter Estimation and Prediction

Parameter estimation in the GPR model—i.e., the coefficients β, noise variance σ^2, and kernel hyperparameters θ—is commonly performed by maximizing the marginal likelihood:

$$p(y \mid X, \beta, \theta, \sigma^2) = \mathcal{N}(y \mid H\beta, K(X, X \mid \theta) + \sigma^2 I) \tag{3.27}$$

The estimation process involves iterative optimization: first, an estimate $\hat{\beta}$ is obtained that maximizes the likelihood for fixed values of θ and σ^2; then, the *profile likelihood* (with β optimized) is maximized with respect to θ and σ^2.

To make predictions at a new input x_{new} given the training data (X, y), the goal is to determine the predictive distribution:

$$p(y_{\text{new}} \mid y, X, x_{\text{new}}) \tag{3.28}$$

which is also Gaussian, with mean and variance derived from the estimated hyperparameters and the kernel function.

The predictive mean (expected value of the prediction y_{new}) is given by:

$$\mathbb{E}[y_{\text{new}} \mid y, X, x_{\text{new}}, \beta, \theta, \sigma^2] = h(x_{\text{new}})^\top \beta + K(x_{\text{new}}^\top, X \mid \theta)\alpha \tag{3.29}$$

where:

$$\alpha = \left(K(X, X \mid \theta) + \sigma^2 I\right)^{-1} (y - H\beta) \tag{3.30}$$

The predictive variance is:

$$\Sigma = k(x_{\text{new}}, x_{\text{new}}) - K(x_{\text{new}}^\top, X) \left(K(X, X) + \sigma^2 I\right)^{-1} K(X, x_{\text{new}}^\top) \tag{3.31}$$

This function-space formulation, enabled by the use of kernels, is computationally advantageous as it avoids explicit operations in the potentially high-dimensional weight space when implicit basis functions are used. Being a probabilistic model, GPR also allows quantification of predictive uncertainty through prediction intervals.

3.7 Forecasting using Vasicek dynamics

The Vasicek model, formally known as the Ornstein-Uhlenbeck (OU) process, is a fundamental single-factor stochastic model used to forecast the evolution of the short-term interest rate, $S(t)$. It is a key mathematical dynamics in finance describing the evolution of interest rates. The model is characterized by having a mean-reverting property, ensuring that the rate tends to revert to a long-run equilibrium level.

3.7.1 Stochastic Differential Equation (SDE)

The dynamics of the short rate $S(t)$ is governed by the following stochastic differential equation (SDE):

$$\mathrm{d}S(t) = \kappa(\theta - S(t))\mathrm{d}t + \sigma \mathrm{d}W(t)$$

where:

- $S(t)$: the short-term interest rate at time t.
- $\kappa > 0$: the mean reversion rate (speed of reversion). This parameter dictates the speed of adjustment towards the long-term mean. A higher κ implies a faster reversion.
- θ: the long-term mean level. This is the equilibrium level of the short rate; all future trajectories of $S(t)$ will evolve around this level in the long run.

- $\sigma > 0$: the instantaneous volatility. This measures instant by instant the amplitude of randomness entering the system, representing the magnitude of random fluctuations. A higher σ implies more randomness.
- $\mathrm{d}W(t)$: the increment of a standard Wiener process (Brownian motion) under the risk-neutral framework, modelling the continuous inflow of randomness into the system.

3.7.2 Exact solution and simulation

The Vasicek model has an exact solution. Given the current rate $S(t)$, the rate at time $t + \Delta t$ follows a normal distribution:

$$S(t + \Delta t) \mid S(t) \sim \mathcal{N}\left(\mu_{\text{cond}}, \sigma^2_{\text{cond}}\right)$$

where the conditional moments are:

- conditional mean:

$$\mu_{\text{cond}} = S(t)e^{-\kappa\Delta t} + \theta(1 - e^{-\kappa\Delta t})$$

- conditional variance:

$$\sigma^2_{\text{cond}} = \frac{\sigma^2}{2\kappa}(1 - e^{-2\kappa\Delta t})$$

These moments show that the process is Gaussian (normally distributed). Historically, the non-zero probability of negative interest rates was considered a limitation of the model. However, this feature proved to be advantageous during the recent prolonged period of negative interest rates (e.g., in the Eurozone), allowing the model to fit market data where strictly positive models (such as Cox-Ingersoll-Ross) struggled.

3.7.3 Parameter Estimation

To implement the Vasicek model effectively for forecasting or pricing, the parameters κ, θ, and σ must be estimated from historical data. Two common statistical approaches for this estimation are the Ordinary Least Squares (OLS) method and the Maximum Likelihood Estimation (MLE).

Discretization

The exact solution implies that the relationship between consecutive observations separated by a time step Δt can be written as a discrete-time autoregressive process of order 1, i.e., AR(1):

$$S_{t+\Delta t} = c + \beta S_t + \varepsilon_t \tag{3.32}$$

where ε_t is a random error term with variance η^2. The relationships between the regression coefficients and the Vasicek parameters are:

$$\beta = e^{-\kappa \Delta t} \tag{3.33}$$

$$c = \theta(1 - e^{-\kappa \Delta t}) \tag{3.34}$$

$$\eta^2 = \frac{\sigma^2}{2\kappa}(1 - e^{-2\kappa \Delta t}) \tag{3.35}$$

Method 1: Ordinary Least Squares (OLS)

The OLS approach treats Eq. (3.32) as a linear regression. By estimating the slope ($\hat{\beta}$), the intercept ($\hat{c}$), and the standard deviation of residuals ($\hat{\eta}$), the parameters are recovered via inversion:

$$\hat{\kappa} = -\frac{\ln(\hat{\beta})}{\Delta t} \tag{3.36}$$

$$\hat{\theta} = \frac{\hat{c}}{1 - \hat{\beta}} \tag{3.37}$$

$$\hat{\sigma} = \hat{\eta} \sqrt{\frac{-2\ln(\hat{\beta})}{\Delta t(1 - \hat{\beta}^2)}} \tag{3.38}$$

Method 2: Maximum Likelihood Estimation (MLE)

The MLE method maximizes the probability that the historical data was generated by the model. Given the Gaussian nature of the process, the log-likelihood function for a sample of n observations is:

$$\mathcal{L}(\kappa, \theta, \sigma) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\sigma^2_{\text{cond}}) - \frac{1}{2\sigma^2_{\text{cond}}} \sum_{i=0}^{n-1} (S_{i+1} - \mu_{\text{cond}}(S_i))^2 \tag{3.39}$$

The optimal parameters are found by maximizing this function using numerical optimization algorithms.

3.7.4 Zero-coupon bond pricing

Once the parameters have been estimated, the Vasicek framework allows for a closed-form solution for the price of a zero-coupon bond. The price at time t of a zero-coupon bond of maturity T is given by:

$$P(t, T) = A(t, T) \cdot e^{-B(t,T)r(t)} \tag{3.40}$$

where $r(t) \equiv S(t)$ is the short-term interest rate, and the functions $A(t, T)$ and $B(t, T)$ depend on the estimated parameters as follows:

$$B(t, T) = \frac{1}{\kappa}\left[1 - e^{-\kappa(T-t)}\right]$$

where:

- $B(t, T)$: the function determining the rate's impact on the bond price.
- T: the bond's maturity time.
- t: the current time.
- κ: the mean reversion rate.

$$A(t, T) = \exp\left\{\left(\theta - \frac{\sigma^2}{2\kappa^2}\right)[B(t, T) - (T - t)] - \frac{\sigma^2 B(t, T)^2}{4\kappa}\right\}$$

where:

- $A(t, T)$: the function accounting for the mean-reversion and volatility of the process.
- θ: the long-term mean level.
- σ: the instantaneous volatility.
- κ: the mean reversion rate.

4 Applications

Having concluded the presentation of the methodologies to be employed, we can now move on to the case studies in which these methods have been applied. To ensure the robustness of our analysis, we replicated our study across different contexts: Main Western Countries, Main Pacific Economies, and Sovereign Yield Curves in European Countries. Each context introduces specific features and distinctive elements, allowing us to build a comprehensive framework that illustrates the performance of the proposed approach on yield curves of different types and across diverse market environments.

We begin with the Main Western Countries case study, which examines the bootstrapped yield curves of major Western economies. These markets are generally characterized by low volatility and similar monetary policies among central banks. However, the selected time range also includes the COVID-19 crisis and the subsequent interest rate hikes, providing an opportunity to observe model behavior under conditions of high volatility.

We then proceed to a similar analysis focusing on the Main Pacific Economies, a more heterogeneous sample composed of currencies from countries in the Pacific area. This group includes large economies such as the United States, China, and Japan, as well as emerging markets such as India, Malaysia, and Taiwan. Their yield curves exhibit markedly different characteristics, as expected from countries with diverse economic structures and unaligned monetary policies. This setting allows us to further test the robustness of our approach across highly differentiated market contexts, avoiding results that could be biased by the structural similarity of the curves. In this second study, we also introduce a another optimization algorithm, Simulated Annealing (SA), which is compared with the two algorithms already used in the previous analysis, namely Genetic Algorithm (GA) and Particle Swarm Optimization (PSO).

Finally, we turn to the third case study, Sovereign Yield Curves in European Countries. This part of the analysis stems from the intention to move beyond bootstrapped curves derived from listed financial instruments, and instead to rely directly on the observed sovereign yield curves of major euro area countries. These curves have specific characteristics, including lower volatility and a less granular maturity structure. In this final study, having already obtained insights from the previous analyses, we not only validate our approach but also explore its practical applications. We demonstrate how the benefits achieved through the improved performance of enhanced parametric models,

made possible by the inclusion of global optimization solvers, can be leveraged to forecast sovereign yield curves using the Vasicek model, calibrated with zero rates derived from the fitted enhanced parametric models.

4.1 Main Western Countries

4.1.1 Data

This study employs the Nelson-Siegel, Svensson, and De Rezende-Ferreira Ferreira models to analyse the following five currencies, Swiss Franc (CHF), Euro (EUR), Japanese Yen (JPY), British Pound (GBP), and US Dollar (USD), over the period from February 1st, 2018 to June 30th, 2023. Traditional mathematical formulas for curve stripping are used for estimating zero rates as well as discount factors for each currency. Table 4.1 provides information on granulometry and instrument types employed in this process. To ensure reliable estimates of the risk-free interest rates term structures, we excluded maturities having more than ten unavailable securities during the bootstrap process. In case some maturity has less than ten missing securities, interpolation methods are applied.

Table 4.1: Granulometry and securities used for the zero rates bootstrap (data collected as of February 1st, 2018)

CHF		EUR		GBP		JPY		USD	
Term	**Instrument Type**	**Term**	**Instrument Type**	**Term**	**Instrument Type**	**Term**	**Instrument Type**	**Term**	**Instrument Type**
1 WK	Swap	6 MO	Deposit	1 WK	Swap	1 WK	Swap	3M	Deposit
2 WK	Swap	EUR 1x7	FRA	2 WK	Swap	2 WK	Swap	FUT_1	Futures
1 MO	Swap	EUR 2x8	FRA	1 MO	Swap	3 WK	Swap	FUT_2	Futures
2 MO	Swap	EUR 3x9	FRA	2 MO	Swap	1 MO	Swap	FUT_3	Futures
3 MO	Swap	EUR 4x11	FRA	3 MO	Swap	2 MO	Swap	FUT_4	Futures
4 MO	Swap	EUR 5x11	FRA	4 MO	Swap	3 MO	Swap	FUT_5	Futures
5 MO	Swap	EUR 6x12	FRA	5 MO	Swap	4 MO	Swap	2Y	Swap
6 MO	Swap	EUR 9x15	FRA	6 MO	Swap	5 MO	Swap	3Y	Swap
7 MO	Swap	EUR 12x18	FRA	7 MO	Swap	6 MO	Swap	4Y	Swap
8 MO	Swap	2 YR	Swap	8 MO	Swap	7 MO	Swap	5Y	Swap
9 MO	Swap	3 YR	Swap	9 MO	Swap	8 MO	Swap	6Y	Swap
10 MO	Swap	4 YR	Swap	10 MO	Swap	9 MO	Swap	7Y	Swap
11 MO	Swap	5 YR	Swap	11 MO	Swap	10 MO	Swap	8Y	Swap
12 MO	Swap	6 YR	Swap	1 YR	Swap	11 MO	Swap	9Y	Swap
18 MO	Swap	7 YR	Swap	18 MO	Swap	12 MO	Swap	10Y	Swap
2 YR	Swap	8 YR	Swap	2 YR	Swap	15 MO	Swap	11Y	Swap
3 YR	Swap	9 YR	Swap	3 YR	Swap	18 MO	Swap	12Y	Swap
4 YR	Swap	10 YR	Swap	4 YR	Swap	2 YR	Swap	15Y	Swap
5 YR	Swap	11 YR	Swap	5 YR	Swap	3 YR	Swap	20Y	Swap
6 YR	Swap	12 YR	Swap	6 YR	Swap	4 YR	Swap	25Y	Swap
7 YR	Swap	15 YR	Swap	7 YR	Swap	5 YR	Swap	30Y	Swap
8 YR	Swap	20 YR	Swap	8 YR	Swap	6 YR	Swap	40Y	Swap
9 YR	Swap	25 YR	Swap	9 YR	Swap	7 YR	Swap	50Y	Swap
10 YR	Swap	30 YR	Swap	10 YR	Swap	8 YR	Swap		
12 YR	Swap	40 YR	Swap	12 YR	Swap	9 YR	Swap		
15 YR	Swap	50 YR	Swap	15 YR	Swap	10 YR	Swap		
20 YR	Swap			20 YR	Swap	11 YR	Swap		
25 YR	Swap			25 YR	Swap	12 YR	Swap		
30 YR	Swap			30 YR	Swap	15 YR	Swap		
						40 YR	Swap		
						50 YR	Swap		

4.1.2 Results

The results clearly highlight that the performance of parametric models for interest rates curve estimation strongly depends on the choice of initial starting points. This sensitivity significantly affects the stability and

reliability of the curve-fitting process, especially when dealing with heterogeneous yield curve shapes such as upward-sloping, flat, humped, or inverted structures. Without the use of artificial intelligence, simpler models with fewer parameters are generally more robust and able to deliver acceptable results even when suboptimal initial values are used. The Nelson-Siegel model, in particular, appears to be the least affected by initialization, showing a higher success rate in achieving convergence across a variety of market conditions. By contrast, more complex models such as Svensson and De Rezende-Ferreira, which involve a larger number of parameters, tend to suffer from poor initialization. In these cases, unsuitable starting values often result in poor performance or even non-convergence. This is clearly visible in Figure 4.1, where Svensson and De Rezende-Ferreira exhibit markedly lower performance compared to Nelson-Siegel in their baseline implementations.

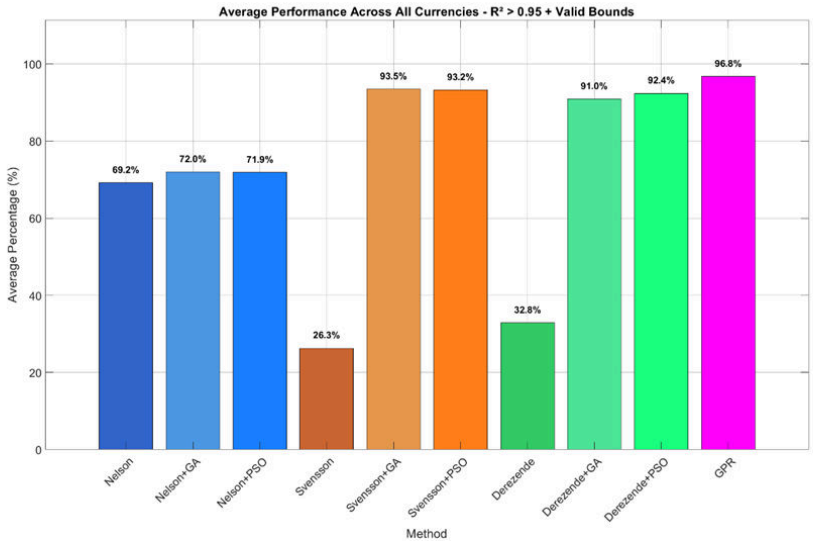


Figure 4.1: Average of fitting performance for different currencies

When optimization is enhanced through the use of Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), these limitations are effectively mitigated. The AI-based methods allow for a broader and more efficient exploration of the parameter space, thereby improving convergence rates and reducing the negative impact of initialization. As illustrated in Figure 4.1, once supported by AI, both the Svensson and De Rezende-Ferreira models achieve average success rates above 90%, surpassing Nelson-Siegel, which remains around 70% even with AI assistance. This pattern is consistent across all currencies, though with varying magnitudes. The most striking cases are observed in CHF, EUR, and GBP, as reported in Figure 4.1, where the performance gap between baseline and AI-enhanced models is particularly significant.

Regarding Gaussian Process Regression (GPR), the results show that, on average, this Machine Learning approach delivers higher performance than all parametric models, including those reinforced by AI. GPR achieves an average success rate of 96.8%, as shown in Figure 4.1. However, Table 4.2 also reveals the presence of specific outliers where

AI-enhanced parametric models perform at least as well as GPR. Notable examples include USD with De Rezende-Ferreira + PSO, USD with Svensson + GA, and GBP with Svensson (both GA and PSO), where the parametric models equal or even surpass the GPR performance.

Table 4.2: Cumulative percentage of curves with $R^2 > 0.95$ and valid bounds by method and currency

Currency	Nelson	Nelson+GA	Nelson+PSO	Svensson	Svensson+GA	Svensson+PSO	De Rezende-Ferreira	De Rezende-Ferreira+GA	De Rezende-Ferreira+PSO	GPR
CHF	65.7	70.4	70.5	0.0	85.6	85.0	0.0	77.3	83.7	97.3
EUR	66.4	70.1	70.2	21.3	95.0	95.0	15.2	95.7	95.4	100.0
GBP	60.7	62.4	60.9	14.3	87.8	87.9	23.6	83.1	84.5	87.7
JPY	99.2	99.3	99.3	87.9	99.3	99.3	42.6	99.3	98.3	99.6
USD	54.1	57.7	58.7	7.9	99.6	99.0	82.8	99.4	100.0	99.4
Average	69.2	72.0	71.9	26.3	93.5	93.2	32.8	91.0	92.4	96.8

Figure 4.2 and Figure 4.3 the RMSE time series obtained using the GPR method for five currencies—CHF, EUR, GBP, JPY, and USD. Figure 4.2 shows the in-sample RMSE, reflecting model accuracy on the training data, while Figure 4.3 displays the out-of-sample RMSE, indicating predictive performance on unseen data.

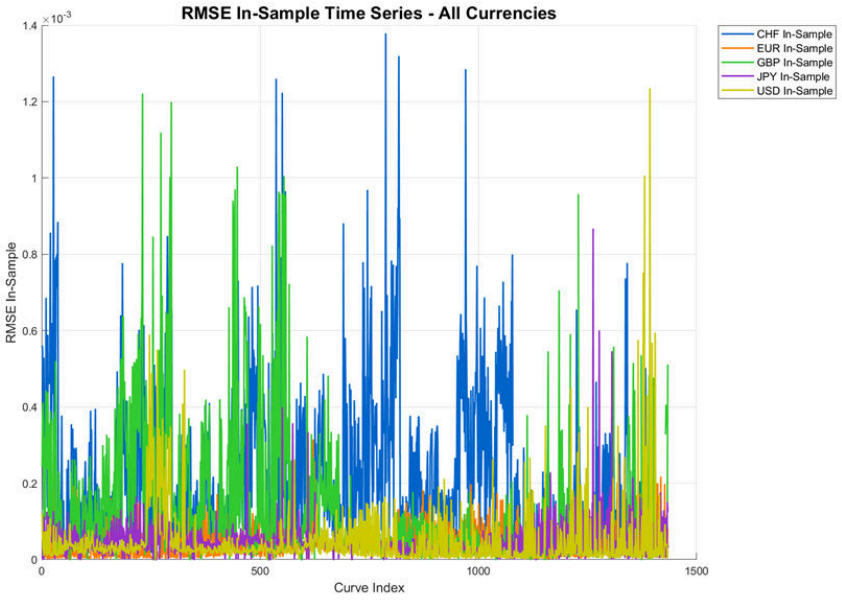


Figure 4.2: In-Sample RMSE Time Series for CHF, EUR, GBP, JPY, and USD

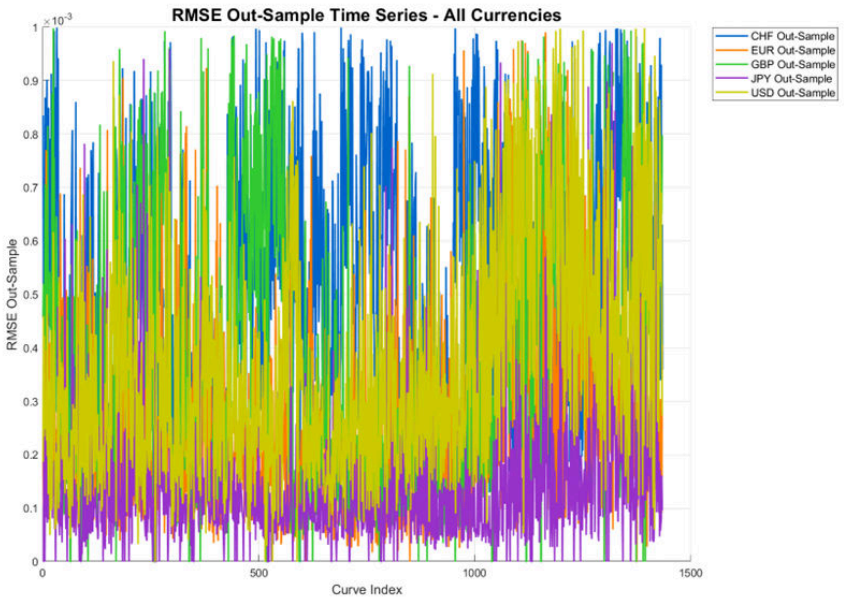


Figure 4.3: Out of-Sample RMSE Time Series for CHF, EUR, GBP, JPY, and USD

Tables 4.3 provides a summary of the results, reporting both absolute and average values for each currency.

Table 4.3: Fitting statistics for different currencies using a Machine Learning approach

Currency	Resolved $R^2 > 0.95$ (%)	Mean RMSE In	Std RMSE In	Mean RMSE Out	Std RMSE Out	Mean R^2	Std R^2
CHF	97.3	0.00020891	0.00020362	0.00049399	0.00024375	0.97746	0.095759
EUR	100.0	0.362	0.411	0.00025567	0.00018344	0.98184	0.12269
GBP	87.7	0.00014494	0.00001782	0.00038475	0.00002683	0.95583	0.17350
JPY	99.6	0.505	0.487	0.00014349	0.00012781	0.95215	0.20631
USD	99.4	0.473	0.0818	0.00036561	0.00021311	0.98844	0.05362

4.2 Main Pacific Economies

4.2.1 Data

This section addresses the Nelson-Siegel, Svensson, and De Rezende-Ferreira Ferreira models to analyse the following nine currencies: Australian Dollar (AUD), Chinese Yuan (CNY), Hong Kong Dollar (HKD), Indian Rupee (INR), Japanese Yen (JPY), Malaysian Ringgit (MYR), New Zealand Dollar (NZD), New Taiwan Dollar (TWD), and United States Dollar (USD). Figure 4.4, The analysis covers the period from January 1st, 2018 to December 12th, 2024.

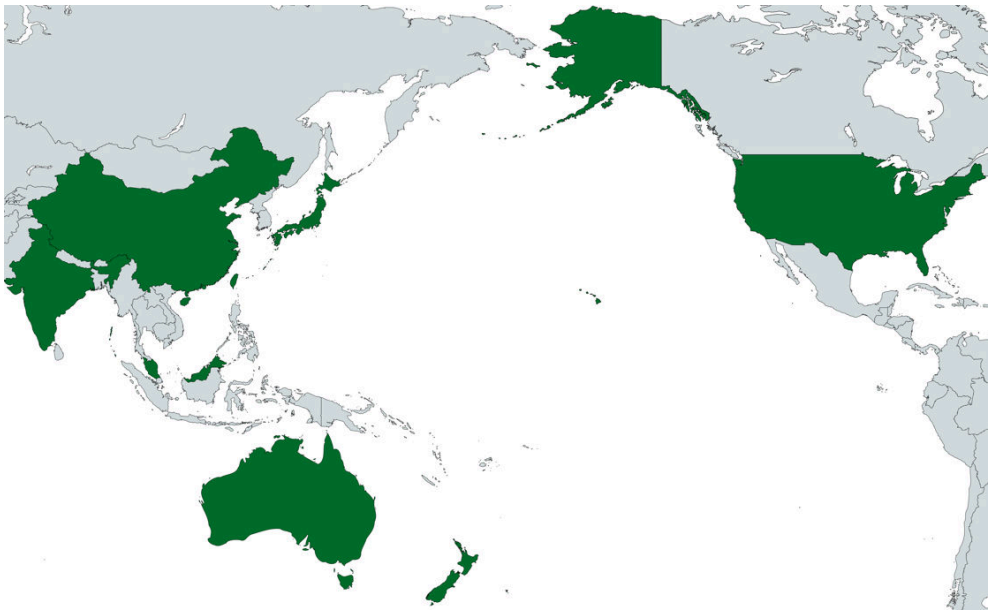


Figure 4.4: The analyzed Pacific area countries

Traditional mathematical procedures for curve stripping are applied to estimate the zero-coupon yield curves and discount factors for each currency, using a wide set of listed securities. Figure 4.5 provides details on the granulometry and the instruments used in the bootstrap process.

USD		AUD		CNY		HKD		INR		MYR		NZD		TWD		JPY	
Term	Instrument	Term	Instrument	Term	Instrument	Term	Instrument	Term	Instrument	Term	Instrument	Term	Instrument	Term	Instrument	Term	Instrument
7 days	Swap	7 days	Deposit	7 days	Deposit	6 months	Swap	1 month	Swap	3 months	Deposit	1 month	Swap	3 months	Deposit	7 days	Swap
14 days	Swap	1 month	Deposit	1 month	Swap	9 months	Swap	2 months	Swap	1 years	Swap	2 months	Swap	1 years	Swap	14 days	Swap
21 days	Swap	2 months	Deposit	3 months	Swap	12 months	Swap	3 months	Swap	2 years	Swap	3 months	Swap	2 years	Swap	21 days	Swap
1 month	Swap	3 months	Swap	6 months	Swap	2 years	Swap	6 months	Swap	3 years	Swap	4 months	Swap	3 years	Swap	1 month	Swap
2 months	Swap	4 months	Swap	9 months	Swap	3 years	Swap	9 months	Swap	4 years	Swap	5 months	Swap	4 years	Swap	2 months	Swap
3 months	Swap	5 months	Swap	12 months	Swap	4 years	Swap	12 months	Swap	5 years	Swap	6 months	Swap	5 years	Swap	3 months	Swap
4 months	Swap	6 months	Swap	1 anno	Swap	5 years	Swap	2 years	Swap	7 years	Swap	9 months	Swap	7 years	Swap	4 months	Swap
5 months	Swap	9 months	Swap	2 years	Swap	7 years	Swap	3 years	Swap	10 years	Swap	1 years	Swap	10 years	Swap	5 months	Swap
6 months	Swap	12 months	Swap	3 years	Swap	10 years	Swap	4 years	Swap	12 years	Swap	2 years	Swap			6 months	Swap
7 months	Swap	18 months	Swap	4 years	Swap	12 years	Swap	5 years	Swap	15 years	Swap	3 years	Swap			7 months	Swap
8 months	Swap	2 years	Swap	5 years	Swap	15 years	Swap	7 years	Swap	20 years	Swap	4 years	Swap			8 months	Swap
9 months	Swap	3 years	Swap	7 years	Swap			10 years	Swap			5 years	Swap			9 months	Swap
10 months	Swap	4 years	Swap	10 years	Swap							6 years	Swap			10 months	Swap
11 months	Swap	5 years	Swap									7 years	Swap			11 months	Swap
12 months	Swap	6 years	Swap									8 years	Swap			12 months	Swap
2 years	Swap	7 years	Swap									9 years	Swap			15 months	Swap
3 years	Swap	8 years	Swap									10 years	Swap			18 months	Swap
4 years	Swap	9 years	Swap									12 years	Swap			2 years	Swap
5 years	Swap	10 years	Swap									15 years	Swap			3 years	Swap
6 years	Swap	12 years	Swap									20 years	Swap			4 years	Swap
7 years	Swap	15 years	Swap									25 years	Swap			5 years	Swap
8 years	Swap	20 years	Swap									30 years	Swap			6 years	Swap
9 years	Swap	25 years	Swap													7 years	Swap
10 years	Swap	30 years	Swap													8 years	Swap
12 years	Swap															9 years	Swap
15 years	Swap															10 years	Swap
20 years	Swap															12 years	Swap
25 years	Swap															15 years	Swap
30 years	Swap															20 years	Swap
35 years	Swap															25 years	Swap
40 years	Swap															30 years	Swap
50 years	Swap															35 years	Swap
																40 years	Swap

Figure 4.5: Granulometry and securities used for the zero rates bootstrap

4.2.2 Results

After fitting the bootstrapped curves with the Nelson-Siegel, Svensson, and De Rezende-Ferreira Ferreira models aided by the Levenberg-Marquardt algorithm, the performance evaluation of the model's fitting is based on two main criteria:

- **Goodness of fit**: Models with an Adjusted R-squared less than 0.95 are considered poor fits and are discarded.
- **Parameter bounds**: To avoid extensive search ranges that could lead to unrealistic solutions, maximum and minimum values for each parameter were established based on values proposed by the Federal Reserve.

The configuration parameters for the three optimization algorithms are presented in Table 4.4. GA and PSO share identical parameter settings, reflecting their common population-based algorithmic structure. Both algorithms utilize the same maximum iteration limits, population sizes, and convergence tolerances. In contrast, SA employs a fundamentally different algorithmic approach based on single-solution exploration with temperature-controlled acceptance criteria. While SA's parameter configuration differs due to its distinct optimization logic, the values have been carefully selected to ensure computational equivalence and fair comparison with the population-based algorithms.

Table 4.4: Configuration parameters for optimization algorithms

Parameter	Genetic Algorithm (GA)		Particle Swarm Optimization (PSO)		Simulated Annealing (SA)	
	Parameter Name	Value	Parameter Name	Value	Parameter Name	Value
Max number of iterations	MaxGenerations	1000	MaxIterations	1000	MaxIterations	1000
Number of candidate solutions	PopulationSize	50	SwarmSize	50		1
Algorithm termination tolerance	FunctionTolerance	1.00×10^{-10}	FunctionTolerance	1.00×10^{-10}	FunctionTolerance	1.00×10^{-8}
Maximum function evaluations		~50,000		~50,000	MaxFunctionEvaluations	$5000 \times nvar$
Initial temperature					InitialTemperature	100
Reanneal interval					ReannealInterval	50
Temperature function					TemperatureFcn	@temperaturefast

The parameter values listed in Table 4.4 were selected to ensure:

- Sufficient exploration of the solution space through adequate population sizes (GA and PSO) or iteration counts (SA)
- Convergence precision through appropriate tolerance thresholds
- Computational efficiency by limiting maximum function evaluations
- Fair comparison across algorithms despite their different optimization strategies

To provide an additional robustness check and strengthen the empirical analysis, we performed both in-sample and out-of-sample RMSE evaluations for all the methodologies considered and across all the currencies included in the study. The analysis shows that RMSE values are consistently low and exhibit temporal correlation patterns. As expected from sound statistical practice, out-of-sample RMSE values

are systematically slightly higher than their in-sample counterparts, confirming the models' ability to generalize beyond the training data without significant overfitting.

The performance evaluation was conducted using the three parametric models—Nelson-Siegel (NS), Svensson (SV), and De Rezende-Ferreira (DR)—each combined with one of four optimization approaches: the baseline Levenberg-Marquardt algorithm and three global optimization algorithms (Genetic Algorithm - GA, Particle Swarm Optimization - PSO, and Simulated Annealing - SA). These global algorithms were used to explore the search space and identify promising regions before launching the local optimization routine, improving convergence and reducing the likelihood of trapping in local minima.

The only currency for which we could not complete the full analysis was the Taiwan Dollar (TWD), as we only had 8 maturities available. In the case of the De Rezende-Ferreira model, this constraint prevented us from conducting the out-of-sample test, since the De Rezende-Ferreira specification requires 8 parameters and this did not allow us to remove additional points from the yield curve to perform the analysis while maintaining statistical validity.

Table 4.5: In-sample RMSE values for yield curve fitting across currencies and models

Currency	Nelson-Siegel				Svensson				De Rezende-Ferreira			
	Base	GA	PSO	SA	Base	GA	PSO	SA	Base	GA	PSO	SA
USD	0.150141	0.1541	0.149787	0.148634	0.08877	0.11647	0.142671	0.133412	0.078348	0.099303	0.12565	0.110019
AUD	0.058857	0.065388	0.062437	0.081834	0.041797	0.056032	0.05491	0.052574	0.033825	0.042575	0.049581	0.044539
CNY	0.053422	0.053227	0.052391	0.054131	0.040596	0.044858	0.049067	0.046376	0.031038	0.043304	0.049144	0.044087
HKD	0.046946	0.048008	0.049274	0.049058	0.037844	0.040763	0.045499	0.041081	0.035442	0.037694	0.04351	0.035461
INR	0.071648	0.074606	0.075467	0.075644	0.040759	0.069615	0.073895	0.064477	0.035579	0.056899	0.070236	0.051599
MYR	0.029368	0.029552	0.029213	0.028991	0.018023	0.018235	0.020782	0.018215	0.013251	0.016276	0.017653	0.014756
NZD	0.063474	0.065065	0.067447	0.067231	0.031474	0.045137	0.049508	0.047636	0.027145	0.038719	0.047213	0.042009
TWD	0.008491	0.008526	0.008228	0.008222	0.00737	0.007021	0.008608	0.006949	NaN	NaN	NaN	NaN
JPY	0.017171	0.017946	0.017699	0.017633	0.010724	0.010949	0.011566	0.011037	0.005395	0.007702	0.010755	0.009218

Table 4.6: Out-of-sample RMSE values for yield curve fitting across currencies and models

Currency	Nelson-Siegel				Svensson				De Rezende-Ferreira			
	Base	GA	PSO	SA	Base	GA	PSO	SA	Base	GA	PSO	SA
USD	0.186448	0.193019	0.186427	0.186517	0.128654	0.159204	0.173688	0.16613	0.119771	0.136023	0.171843	0.152006
AUD	0.115422	0.113944	0.110469	0.109471	0.060452	0.072115	0.07786	0.072589	0.056995	0.062522	0.072015	0.06855
CNY	0.098902	0.097824	0.099141	0.097131	0.104102	0.105092	0.113594	0.111231	0.116092	0.121489	0.14103	0.119075
HKD	0.121298	0.11806	0.117104	0.116216	0.129888	0.120099	0.111247	0.119372	0.137318	0.130701	0.143074	0.127289
INR	0.140411	0.142718	0.143583	0.146371	0.140214	0.158544	0.163205	0.156901	0.14022	0.214771	0.239907	0.181992
MYR	0.068998	0.068032	0.067853	0.067074	0.074476	0.071113	0.065073	0.070383	0.06677	0.088347	0.089744	0.088599
NZD	0.085113	0.088893	0.087094	0.089306	0.053631	0.067681	0.07167	0.072101	0.056148	0.064734	0.07807	0.068439
TWD	0.037438	0.03431	0.0388	0.04316	0.037882	0.039821	0.035371	0.043019	NaN	NaN	NaN	NaN
JPY	0.024163	0.024437	0.024309	0.023981	0.014489	0.015542	0.017459	0.016743	0.008862	0.011338	0.017219	0.015088

As shown in the Tables 4.5 and 4.6, the RMSE values demonstrate consistent model performance across different optimization approaches. The systematic increase from in-sample to out-of-sample RMSE values indicates healthy model behavior without excessive overfitting. The relatively small differences between in-sample and out-of-sample errors across all currencies (except TWD for the De Rezende-Ferreira model) confirm the robustness and generalization capability of the fitted yield curve models.

Finally, the overall shown in the Tables 4.7 highlight the performance of the various methods are reported in the table below. In addition to the methodologies based on parametric models, we also introduce a machine learning approach through Gaussian Process Regression (GPR),

providing a challenging benchmark against which to compare the results obtained through parametric models aided by nature-inspired algorithms.

Table 4.7: Overall performance comparison: Parametric models with optimization algorithms vs. Gaussian Process Regression

Currency	Base Models			NS + Alg.			SV + Alg.			DR + Alg.			GPR
	NS	SV	DR	GA	PSO	SA	GA	PSO	SA	GA	PSO	SA	
AUD	61.6	33.6	70.1	66.7	66.6	66.6	88.0	87.4	80.7	89.4	88.5	86.9	98.9
USD	43.1	50.3	1.5	42.7	42.7	42.3	56.2	56.0	54.9	58.8	58.7	54.3	99.4
CNY	45.7	5.6	5.2	48.6	48.6	48.6	54.0	54.5	53.7	59.2	58.6	55.0	84.7
HKD	48.3	25.9	22.7	53.6	53.9	53.7	58.9	60.0	58.9	64.7	73.3	64.7	96.9
INR	39.0	9.1	6.0	46.6	46.5	46.7	38.9	44.4	38.9	67.7	70.1	63.6	97.5
MYR	86.0	35.8	33.2	94.3	94.3	94.3	99.6	97.1	97.1	97.1	99.7	99.6	100.0
NZD	67.5	31.4	44.5	74.6	74.6	74.6	98.1	73.9	73.2	74.0	99.3	98.7	100.0
TWD	66.7	37.2	19.3	92.3	92.8	93.2	96.0	95.2	95.1	97.5	97.5	97.0	99.9
JPY	97.5	94.3	37.6	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9
Mean	61.7	35.9	26.7	68.8	68.9	68.9	76.6	74.3	72.5	78.7	82.8	80.0	97.5

The results clearly demonstrate the progressive improvement in fitting performance as we move from baseline parametric models to their enhanced versions incorporating nature-inspired optimization algorithms. The table reveals several important insights:

- **Baseline parametric models**: The basic Nelson-Siegel, Svensson, and De Rezende-Ferreira models show varying performance across currencies, with mean accuracies of 61.7%, 35.9%, and 26.7%, respectively.
- **Enhanced parametric models**: The integration of nature-inspired algorithms (GA, PSO, SA) consistently improves performance across all three parametric model families. The De Rezende-Ferreira model combined with optimization algorithms shows the best performance among parametric approaches, achieving a mean accuracy of 78.7% (DR+GA), 82.8% (DR+PSO), and 80.0% (DR+SA).
- **Machine learning benchmark**: The Gaussian Process Regression (GPR) approach establishes a strong benchmark with an impressive mean accuracy of 97.5%, demonstrating the potential of machine learning techniques in yield curve fitting. But some currencies, such as MYR and NZD, show particularly high performance with the enhanced parametric models, approaching or matching GPR performance. In contrast, currencies like USD and CNY exhibit more challenging fitting characteristics, where the gap between parametric and machine learning approaches remains more pronounced.

These findings suggest that while nature-inspired optimization algorithms significantly enhance the performance of traditional parametric models, machine learning approaches like GPR still offer superior fitting capabilities, particularly for complex yield curve structures. However, the enhanced parametric models provide a valuable middle ground, combining improved accuracy with the interpretability and theoretical foundation of parametric specifications.

The results presented in Figure 4.6 demonstrate a significant transformation in the performance landscape of parametric yield curve models when enhanced with global optimization algorithms. The baseline parametric models initially showed limited effectiveness, with mean success rates of 61.7% for Nelson-Siegel, 35.9% for Svensson, and only 26.7% for De Rezende-Ferreira when fitting curves with $R^2 > 0.95$ and parameters within valid bounds. This poor performance particularly affected the more complex models, which struggled with parameter initialization and convergence issues. The integration of global search algorithms—Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA)—has fundamentally reversed this situation. Enhanced parametric models now achieve substantially improved performance, with mean success rates reaching 68.8–68.9% for Nelson-Siegel variants, 72.5–76.6% for Svensson variants, and 78.7–82.8% for De Rezende-Ferreira variants. These improvements bring parametric models significantly closer to GPR performance (97.5% mean success rate), dramatically narrowing the performance gap that previously existed. Notably, several instances demonstrate where enhanced parametric models either match or exceed GPR performance. For MYR, the De Rezende-Ferreira+PSO model achieves 99.7% success compared to GPR's 100%, while Svensson+GA reaches 99.6%. Similarly, for NZD, both De Rezende-Ferreira+PSO (99.3%) and De Rezende-Ferreira+SA (98.7%) approach GPR's perfect 100% performance. In JPY markets, all enhanced models achieve 99.9% success, effectively matching GPR's performance.

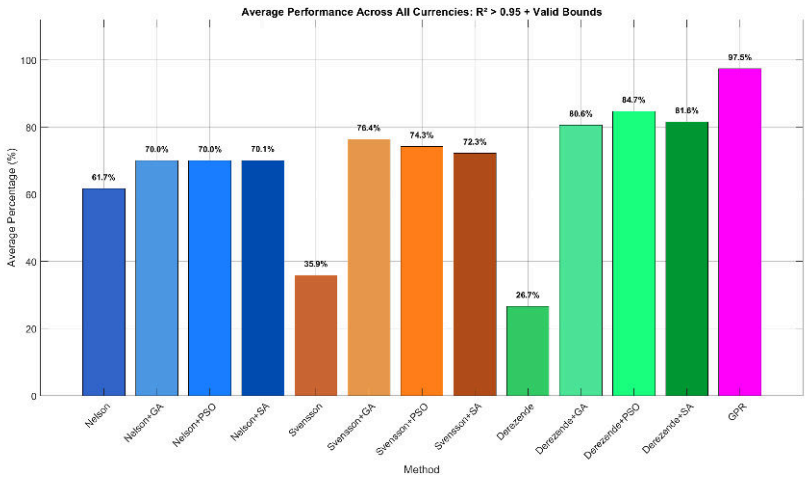


Figure 4.6: Average of fitting performance for different currencies

4.3 Sovereign Yield Curves in European Countries

4.3.1 Data

This section analyses the sovereign yield curves of five major European economies: France, Germany, Italy, Spain, and the United Kingdom Figure 4.7. The analysis covers the period from June 30th, 2015 to June 30th, 2025, considering maturities ranging from three months (3M) to thirty years (30Y), including intermediate terms of 6M, 1Y, 2Y, 3Y, 4Y, 5Y, 6Y, 7Y, 8Y, 9Y, 10Y, 15Y, 20Y, and 25Y.



Figure 4.7: the European countries analysed in the study

Unlike studies focused on zero-coupon curve estimation, this work relies directly on observed government bond yields for the entire maturity spectrum, thus eliminating the need for bootstrap procedures.

4.3.2 Results

The performance evaluation of the Nelson-Siegel (NS), Svensson (SV), and De Rezende-Ferreira (DR) models on the European sovereign curves confirms the significant benefit of incorporating nature-inspired optimization algorithms. The assessment strictly adhered to two criteria: an Adjusted $R^2 \geq 0.95$ (Goodness of Fit) and ensuring all parameters remained within established bounds (Model Robustness).

As illustrated in Figure 4.8, the visual comparison clearly demonstrates how the assistance of optimization algorithms brings a substantial advantage across all three parametric models. The baseline Levenberg-Marquardt approach (Base) shows considerably lower success rates, particularly for the more complex Svensson and De Rezende-Ferreira models. The introduction of nature-inspired algorithms—Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA)—dramatically improves model performance, with success rates consistently exceeding 90% for most configurations.

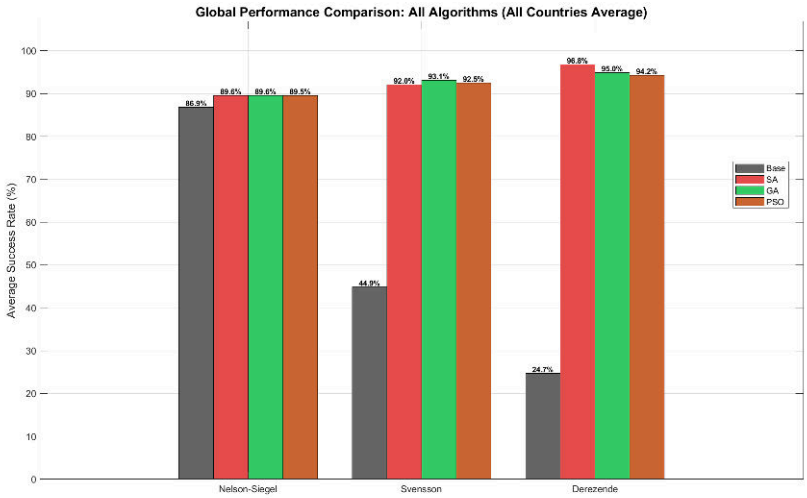


Figure 4.8: Comparative performance of parametric models with and without optimization algorithms

To examine the results in greater detail, Table 4.8 presents the country-specific success rates for each model-algorithm combination. The Svensson model's global success rate nearly doubles from 44.9% (Base) to over 92.0% when aided by GA, PSO, or SA. The De Rezende-Ferreira model shows the most substantial improvement, jumping from a baseline success rate of 24.7% to over 94.2%, highlighting the necessity of global search strategies to locate robust and realistic parameter solutions. Even the simpler Nelson-Siegel model benefits from optimization, with the global mean success rate increasing from 86.9% to approximately 89.6%.

Table 4.8: Performance comparison: Success rate ($R^2 \geq 0.95$ + Parameters within bounds)

Country	Nelson-Siegel				Svensson				De Rezende-Ferreira			
	Base	SA	GA	PSO	Base	SA	GA	PSO	Base	SA	GA	PSO
UK	80.4%	80.5%	80.5%	80.4%	34.5%	84.9%	86.4%	85.1%	36.9%	90.1%	89.0%	87.1%
Spain	90.9%	90.9%	90.9%	90.9%	51.7%	94.4%	95.1%	95.2%	12.4%	99.7%	96.2%	96.5%
Italy	97.1%	97.1%	97.2%	97.2%	54.6%	97.8%	98.1%	97.8%	17.1%	98.4%	98.7%	98.6%
Germany	75.6%	89.1%	89.0%	88.9%	31.0%	91.0%	92.7%	91.6%	38.3%	97.4%	95.6%	94.2%
France	90.3%	90.3%	90.3%	90.3%	52.7%	92.0%	93.1%	92.9%	19.1%	98.5%	95.2%	94.8%
Global Mean	**86.9%**	**89.6%**	**89.6%**	**89.5%**	**44.9%**	**92.0%**	**93.1%**	**92.5%**	**24.7%**	**96.8%**	**95.0%**	**94.2%**

Based on the observed results, in 4 out of 5 cases the best-performing model is De Rezende-Ferreira + SA, with the exception of Italy where De Rezende-Ferreira + GA achieves the highest success rate. This consistency across different sovereign curves demonstrates the robustness and effectiveness of the Simulated Annealing algorithm when combined with the more complex De Rezende-Ferreira parametrization.

However, to verify that the improvements in fitting quality do not come at the cost of excessive computational time, the execution time required by each model-algorithm combination was measured and compared. As illustrated in Figure 4.9, the computational efficiency analysis reveals a remarkable finding: the SA algorithm not only achieves superior performance in 4 out of 5 cases, but is also by far the fastest approach among the three nature-inspired algorithms tested.

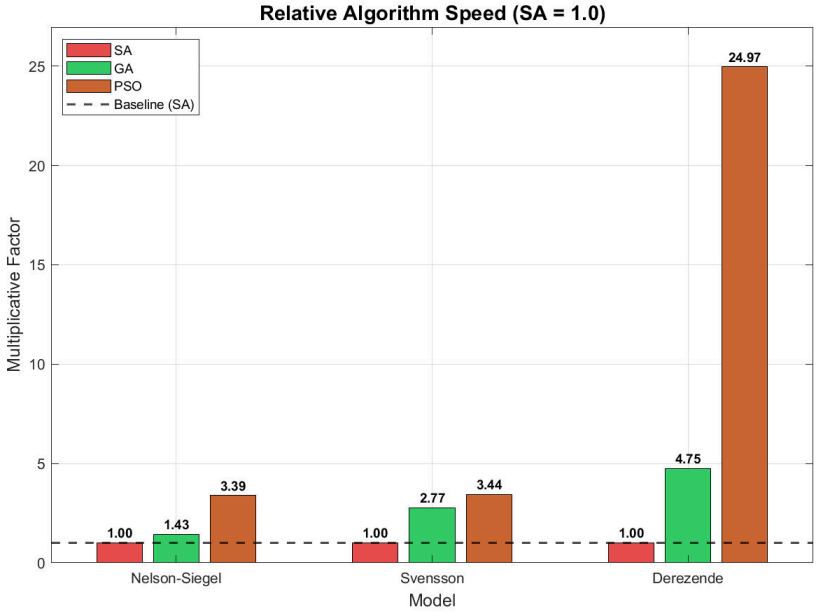


Figure 4.9: Relative computational speed of the different optimization algorithms

The histogram clearly shows that Simulated Annealing significantly outperforms both GA and PSO in terms of computational speed. While GA demonstrates moderate computational requirements, PSO proves to be markedly slower, requiring substantially more processing time to converge to optimal solutions. This stark difference in computational efficiency, combined with SA's superior fitting performance, makes

it the most attractive choice for practical yield curve modeling applications where both accuracy and computational speed are important considerations.

Given these results, we proceed to forecast the yield curve for each country using the Vasicek model, a forecasting model specifically designed for interest rate prediction. Starting from the methods that produced the best results—namely De Rezende-Ferreira + SA for France, Germany, Spain, and the UK, and De Rezende-Ferreira + GA for Italy—we derive the zero rates for each country's curve from 3 months to 30 years, advancing in 3-month intervals. This process yields a total of 120 zero rates for each country over our observation period (30.06.2015 – 30.06.2025).

Using the zero rates corresponding to the 3-month maturity, we estimate the three key parameters of the Vasicek model: the speed of mean reversion (κ), the long-term mean level (θ), and the volatility of the short rate (σ).

Once these parameters are estimated for each country, they are incorporated into the stochastic differential equation governing the Vasicek process to generate multiple potential future interest rate paths. Taking Italy as an example, Figure 4.10 illustrates the 30-year interest rate curve projected for the first forecasting day, July 1st, 2025, based on ten simulated scenarios.

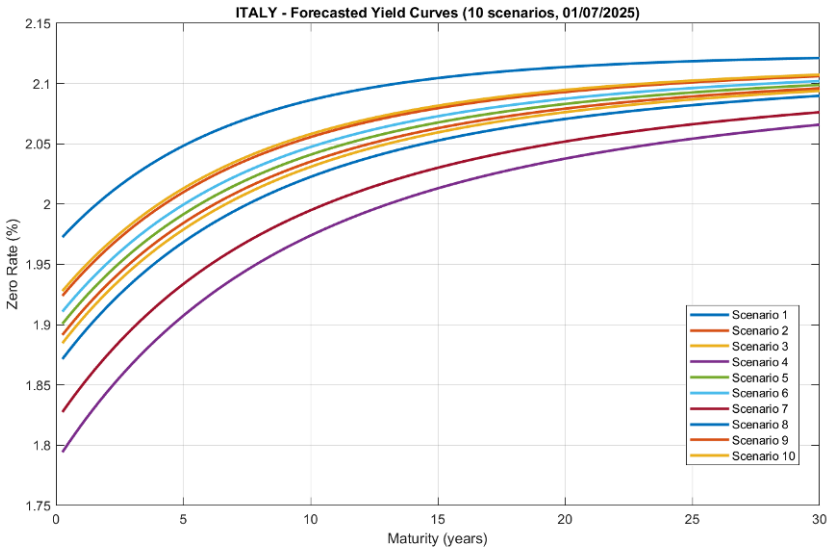


Figure 4.10: Forecast of the 30-year interest rates curve for Italian government bonds on July 1, 2025 (10 simulations)

5 Conclusions

The study demonstrates that the calibration of yield curve models is fundamentally sensitive to the choice of initial parameter values. This dependency directly affects convergence, stability, and reliability, particularly for models with a higher number of parameters. While the Nelson-Siegel model proves to be relatively robust across different market conditions, more complex models such as Svensson and De Rezende-Ferreira initially suffer from poor initialization and often fail to converge when suitable starting points are not available. These findings support a fundamental principle: a higher number of parameters becomes a real advantage only if good starting points can be identified or generated. Otherwise, model complexity turns into a weakness, leading to instability and non-convergence. However, when supported by global optimization techniques such as Genetic Algorithms, Particle Swarm Optimization, and Simulated Annealing, complex models are able to overcome this limitation and fully exploit their flexibility.

The three optimization algorithms demonstrate significant improvements in the fitting performance of parametric yield curve models. Although their performance levels may appear comparable in terms of accuracy, the analysis of European sovereign yield curves reveals important differences in computational efficiency. Specifically, Simulated Annealing proves to be substantially more efficient in terms of computation time compared to the two Artificial Intelligence–based algorithms, GA and PSO.

An additional implication of this result concerns the relationship between model performance and interpretability. Machine Learning techniques, such as Gaussian Process Regression, typically achieve higher accuracy, but at the cost of economic explainability, since they operate as black-box models. By contrast, parametric models maintain a strong link with financial theory and the economic interpretation of the yield curve. The introduction of intelligent optimization allows these models to close the performance gap with Machine Learning approaches while preserving their theoretical transparency.

In conclusion, the research shows that parametric yield curve models, once enhanced through intelligent optimization strategies, achieve a balance between robustness, accuracy, and interpretability. This makes them not only competitive with advanced Machine Learning methods but also preferable in contexts where theoretical consistency and explainability are essential requirements for regulatory compliance and risk management applications.

Finally, regarding possible practical implications, having demonstrated that the integration of global optimization algorithms enhances the fitting capabilities of parametric models, this opens the door to several applied uses. One of these is forecasting, as shown through the Vasicek model application, but the potential applications are numerous. For instance, more accurate and economically consistent fitted curves can be employed for stress testing under different market scenarios, or for the precise valuation and hedging of fixed-income portfolios. Moreover, such improved curve estimation techniques can support the calibration of macro-financial models and the assessment of monetary policy transmission. These analyses would greatly benefit from being based on yield curves calibrated with a high level of precision, yet without sacrificing the economic interpretability and theoretical rigor ensured by parametric modeling.

Bibliography

K. J. Adams and D. R. Van Deventer. Fitting Yield Curves and Forward Rate Curves With Maximum Smoothness. *JFI*, 4(1):52–62, June 1994. ISSN 1059-8596, 2168-8648. doi: 10.3905/jfi.1994.408102. URL http://pm-research.com/lookup/doi/10.3905/jfi.1994.408102.

S. Ahmed, M. M. Alshater, A. E. Ammari, and H. Hammami. Artificial intelligence and machine learning in finance: A bibliometric review. *Research in International Business and Finance*, 61:101646, October 2022. ISSN 02755319. doi: 10.1016/j.ribaf.2022.101646. URL https://linkinghub.elsevier.com/retrieve/pii/S0275531922000344.

J. Annaert, A. G. P. Claes, M. J. K. De Ceuster, and H. Zhang. Estimating the Yield Curve Using the Nelson-Siegel Model: A Ridge Regression Approach. *SSRN Journal*, 2012. ISSN 1556-5068. doi: 10.2139/ssrn.2054689. URL http://www.ssrn.com/abstract=2054689.

S. Bie, F. X. Diebold, J. He, and J. Li. Machine Learning and the Yield Curve: Tree-Based Macroeconomic Regime Switching, May 2025. URL http://arxiv.org/abs/2408.12863. arXiv:2408.12863 [econ].

V. Brousseau. The Functional Form of Yield Curves. *SSRN Journal*, 2002. ISSN 1556-5068. doi: 10.2139/ssrn.314901. URL https://www.ssrn.com/abstract=314901.

A. Cafferata, P. G. Giribone, and M. Resta. *Interest rates term structure models and their impact on actuarial forecasting*. 2018.

A. Cafferata, P. G. Giribone, M. Neffelli, and M. Resta. Yield Curve Estimation Under Extreme Conditions: Do RBF Networks Perform Better? In *Neural Advances in Processing Nonlinear Dynamic Signals*, volume 102, pages 241–251. Springer International Publishing, Cham, 2019. ISBN 978-3-319-95098-3. doi: 10.1007/978-3-319-95098-3_22. Series Title: Smart Innovation, Systems and Technologies.

A. J. G. Cairns and D. J. Pritchard. Stability of models for the term structure of interest rates with application to german market data. *British Actuarial Journal*, 7(3):467–507, 2001. URL http://www.jstor.org/stable/41141518.

P. K. Chu, K. Hoff, P. Molnár, and M. Olsvik. Crude oil: Does the futures price predict the spot price? *Research in International Business and Finance*, 60:101611, April 2022. ISSN 02755319. doi: 10.1016/j.ribaf.2021.101611. URL https://linkinghub.elsevier.com/retrieve/pii/S0275531921002324.

M. Dahlquist and L. E. O. Svensson. Estimating the Term Structure of Interest Rates for Monetary Policy Analysis. *The Scandinavian Journal of Economics*, 98(2):163, June 1996. ISSN 03470520. doi: 10.2307/3440852. URL https://www.jstor.org/stable/3440852?origin=crossref.

R. Dawkins. *The Selfish Gene*. Oxford University Press, Oxford, 1976. URL https://www.jstor.org/stable/10.2307/4381296.

K. A. De Jong. *Analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, Ann Arbor, MI, 1975. URL https://deepblue.lib.umich.edu/handle/2027.42/4507.

R. B. De Rezende and M. S. Ferreira. Journal of Forecasting - 2011 - Rezende - Modeling and Forecasting the Yield Curve by an Extended Nelson-Siegel Class of.pdf, 2013.

M. Deacon and A. J. Derry. *Estimating the term structure of interest rates*. UMI books on demand. UMI, Ann Arbor, facs edition, 1997. ISBN 978-1-85730-082-6.

P. Delucchi and P. G. Giribone. Modeling the interest rates term structure using Machine Learning: a Gaussian process regression approach. *RMM*, 18(3):16–35, December 2023. ISSN 26123665, 27242153. doi: 10.47473/2020rmm0131. URL https://www.aifirm.it/wp-content/uploads/2023/12/RMM-2023-03-Excerpt-2.pdf.

E. F. Fama and R. R. Bliss. The Information in Long-Maturity Forward Rates. 1987.

D. Filipovic, M. Pelger, and Y. Ye. Stripping the Discount Curve - a Robust Machine Learning Approach. *SSRN Journal*, 2022. ISSN 1556-5068. doi: 10.2139/ssrn.4058150. URL https://www.ssrn.com/abstract=4058150. Publisher: Elsevier BV.

M. Fisher, D. Nychka, and D. Zervos. Fitting the Term Structure of Interest Rates with Smoothing Splines. 1994.

P. G. Giribone and O. Caligaris. Modellizzare la curva dei rendimenti mediante metodologie di apprendimento artificiale: analisi e confronto prestazionale tra le tecniche regressive tradizionali e le reti neurali. —, 2015.

R. S. Gürkaynak, B. Sack, and J. H. Wright. The U.S. Treasury yield curve: 1961 to the present. *Journal of Monetary Economics*, 54(8): 2291–2304, November 2007. ISSN 03043932. doi: 10.1016/j.jmoneco.2007.06.029. URL https://linkinghub.elsevier.com/retrieve/pii/S0304393207000840.

P. S. Hagan and G. West. Interpolation Methods for Curve Construction. *Applied Mathematical Finance*, 13(2):89–129, June 2006. ISSN 1350-486X, 1466-4313. doi: 10.1080/13504860500396032. URL http://www.tandfonline.com/doi/abs/10.1080/13504860500396032.

J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, 1975. ISBN 978-0-472-08460-9. Google-Books-ID: JE5RAAAAMAAJ.

J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, Perth, WA, Australia, 1995. IEEE. ISBN 978-0-7803-2768-9. doi: 10.1109/ICNN.1995.488968. URL http://ieeexplore.ieee.org/document/488968/.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598):671–680, May 1983. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.220.4598.671. URL https://www.science.org/doi/10.1126/science.220.4598.671.

K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, 2(2):164–168, July 1944. ISSN 0033-569X, 1552-4485. doi: 10.1090/qam/10666. URL https://www.ams.org/qam/1944-02-02/S0033-569X-1944-10666-0/.

R. B. Litterman and J. Scheinkman. Common Factors Affecting Bond Returns. 1991.

P. Manousopoulos and M. Michalopoulos. Comparison of non-linear optimization algorithms for yield curve estimation. *European Journal of Operational Research*, 192(2):594–602, January 2009. ISSN 03772217. doi: 10.1016/j.ejor.2007.09.017. URL https://linkinghub.elsevier.com/retrieve/pii/S0377221707009381.

D. W. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, June 1963. ISSN 0368-4245, 2168-3484. doi: 10.1137/0111030. URL http://epubs.siam.org/doi/10.1137/0111030.

K. Mastronikola. Yield curves for gilt-edged stocks: a new model, 1991.

J. H. McCulloch. Measuring the Term Structure of Interest Rates. *J BUS*, 44(1):19, January 1971. ISSN 0021-9398, 1537-5374. doi: 10.1086/295329. URL https://www.jstor.org/stable/2351832.

P. Moscato. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts Towards Memetic Algorithms. 1989.

J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313, January 1965. ISSN 0010-4620, 1460-2067. doi: 10.1093/comjnl/7.4.308. URL https://academic.oup.com/comjnl/article-lookup/doi/10.1093/comjnl/7.4.308.

C. R. Nelson and A. F. Siegel. Parsimonious Modeling of Yield Curves. *J BUS*, 60(4):473, January 1987. ISSN 0021-9398, 1537-5374. doi: 10.1086/296409. URL https://www.jstor.org/stable/2352957.

J. Nocedal. Updating Quasi-Newton Matrices with Limited Storage. *Mathematics of Computation*, 35(151):773–782, 1980. doi: 10.1090/S0025-5718-1980-0572855-7.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 3. print edition, 2008. ISBN 978-0-262-18253-9.

G. A. F. Seber. *Nonlinear regression*. Wiley series in probability and mathematical statistics. Wiley, New York, 2004. ISBN 978-0-471-47135-6 978-0-471-72530-5.

G. S. Shea. Pitfalls in Smoothing Interest Rate Term Structure Data: Equilibrium Models and Spline Approximations. *The Journal of Financial and Quantitative Analysis*, 19(3):253, September 1984. ISSN 00221090. doi: 10.2307/2331089. URL https://www.jstor.org/stable/2331089?origin=crossref.

Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pages 69–73, Anchorage, AK, USA, 1998. IEEE. ISBN 978-0-7803-4869-1. doi: 10.1109/ICEC.1998.699146. URL http://ieeexplore.ieee.org/document/699146/.

J. M. Steeley. Estimating the gilt-edged term structure: basis splines and confidence intervals. *Business Finance & Accounting*, 18(4):513–529, June 1991. ISSN 0306-686X, 1468-5957. doi: 10.1111/j.1468-5957.1991.tb01119.x. URL https://onlinelibrary.wiley.com/doi/10.1111/j.1468-5957.1991.tb01119.x.

L. E. O. Svensson. Estimating and Interpreting Forward Interest Rates: Sweden 1992-1994". 1994.

O. A. Vasicek and H. G. Fong. Term Structure Modeling Using Exponential Splines. *The Journal of Finance*, 37(2):339–348, May 1982. ISSN 0022-1082, 1540-6261. doi: 10.1111/j.1540-6261.1982.tb03555.x. URL https://onlinelibrary.wiley.com/doi/10.1111/j.1540-6261.1982.tb03555.x.

D. F. Waggoner. Spline Methods for Extracting Interest Rate Curves from Coupon Bond Prices. *SSRN Journal*, 1998. ISSN 1556-5068. doi: 10.2139/ssrn.86789. URL http://www.ssrn.com/abstract=86789.

W. W. S. Wei. *Time Series Analysis: Univariate and Multivariate Methods*. Redwood City, CA, 1990.

A Appendix

A.1 Main Western Countries

The following figures depict the surfaces of the yield curves for the five currencies examined, each corresponding to a major developed economy. These plots illustrate the term structure of interest rates across different maturities.

A.1.1 IR term structures (Developed Economies)

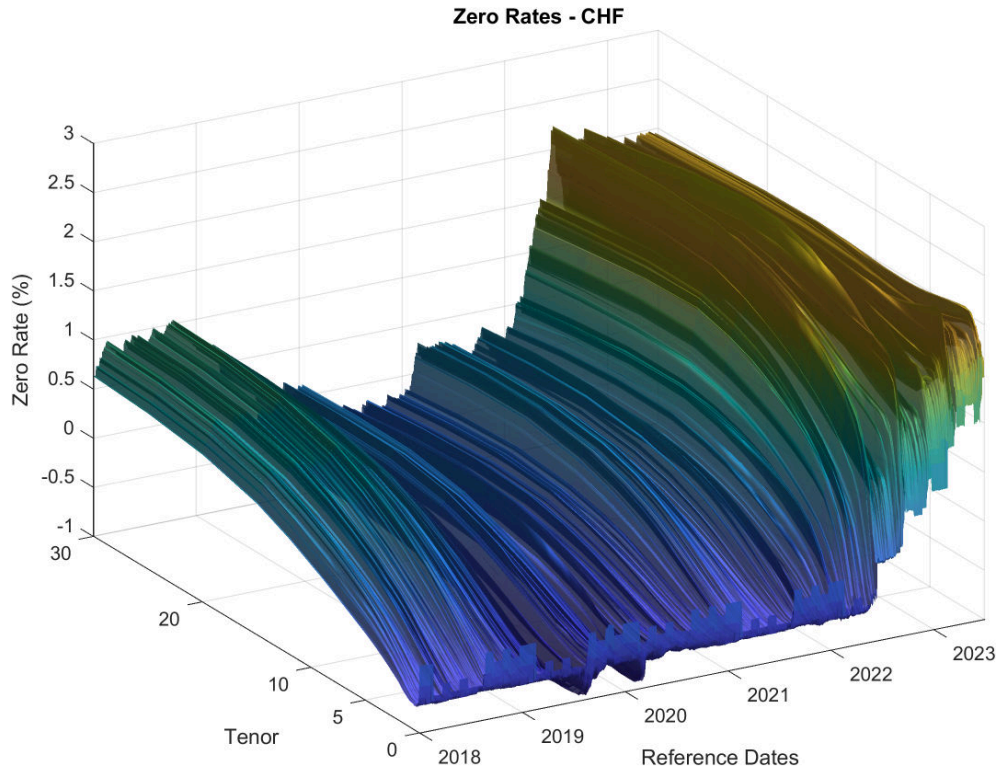


Figure A.1: Interest rates term structure – CHF

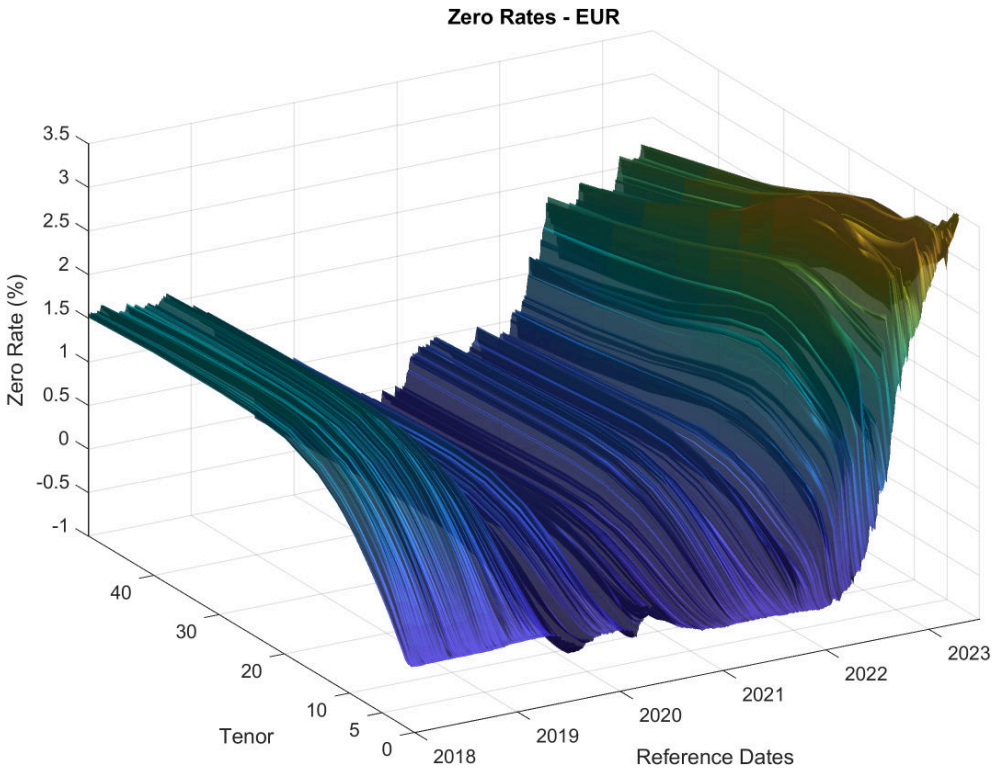


Figure A.2: Interest rates term structure – EUR

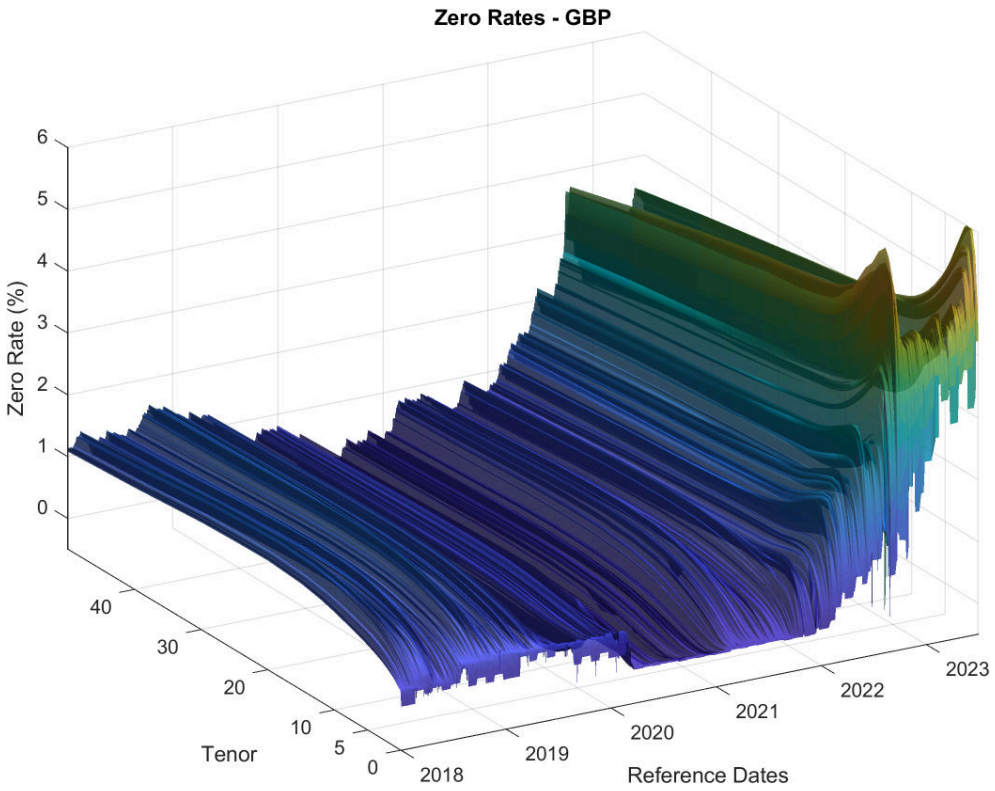


Figure A.3: Interest rates term structure – GBP

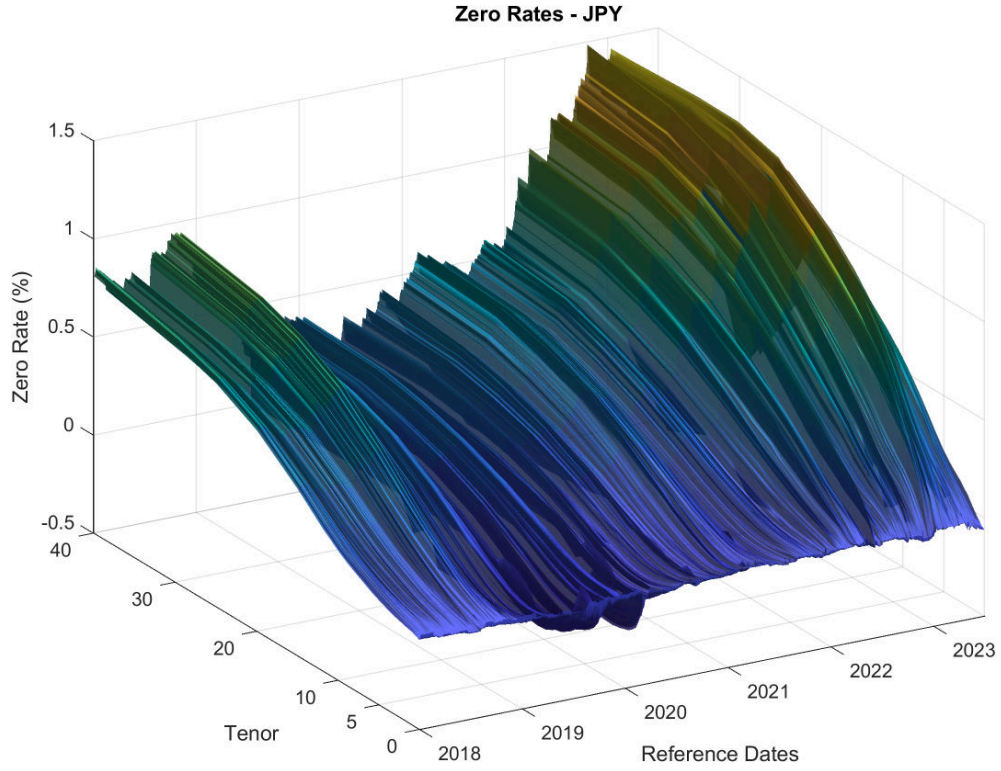


Figure A.4: Interest rates term structure – JPY

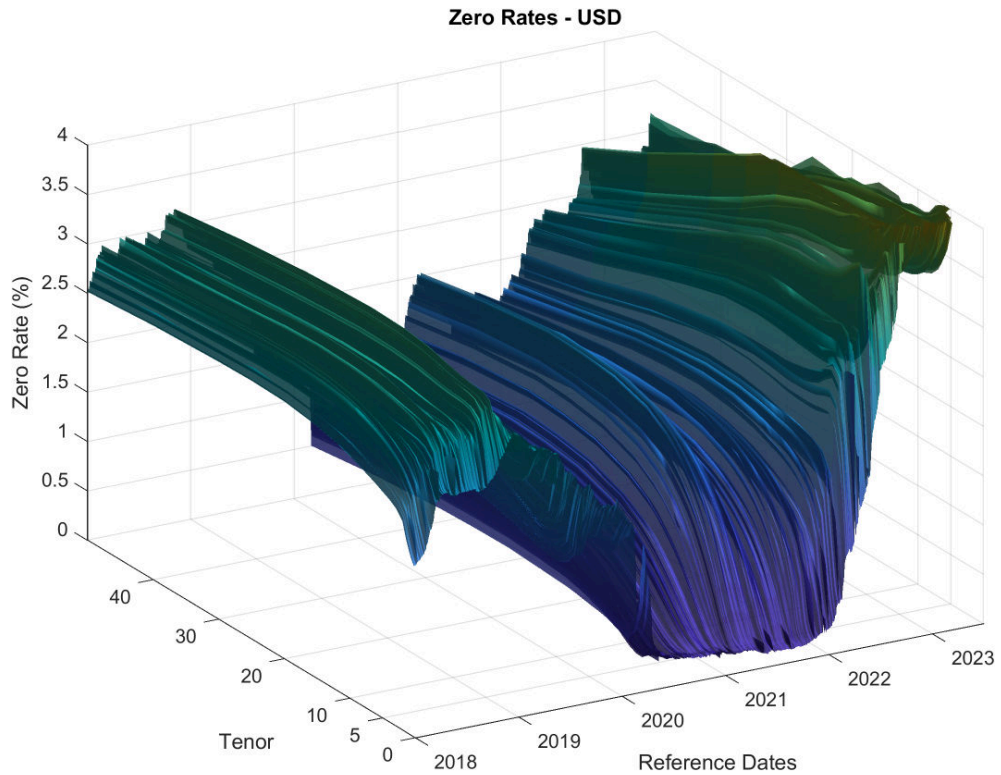


Figure A.5: Interest rates term structure – USD

A.1.2 Base Models (Developed Economies)

The following plots display in blue the estimated coefficients for each parameter, corresponding to the yield curves successfully fitted across the selected currencies.

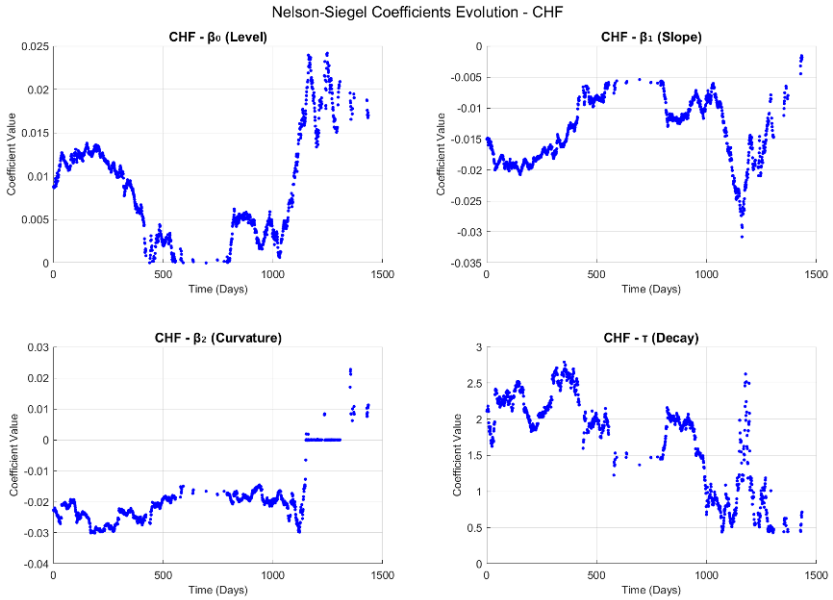


Figure A.6: Nelson-Siegel model coefficients – CHF

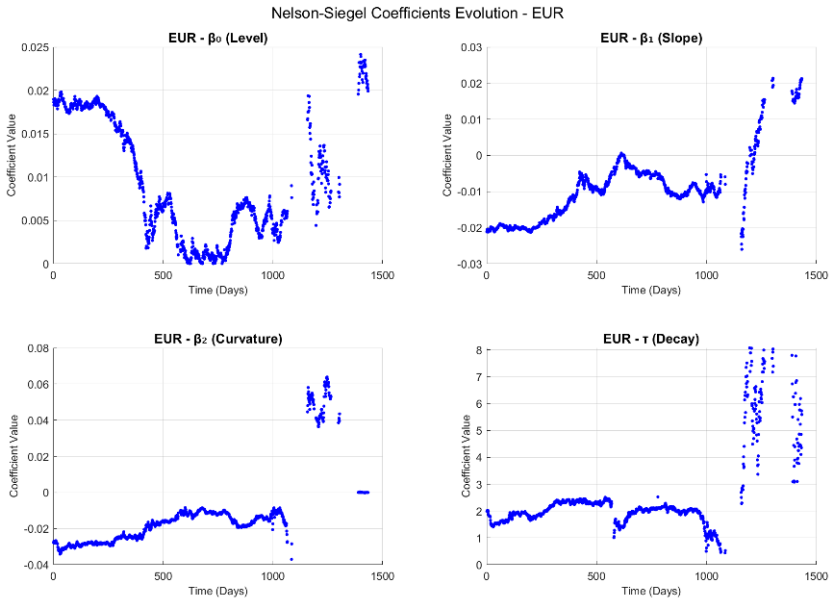


Figure A.7: Nelson-Siegel model coefficients – EUR

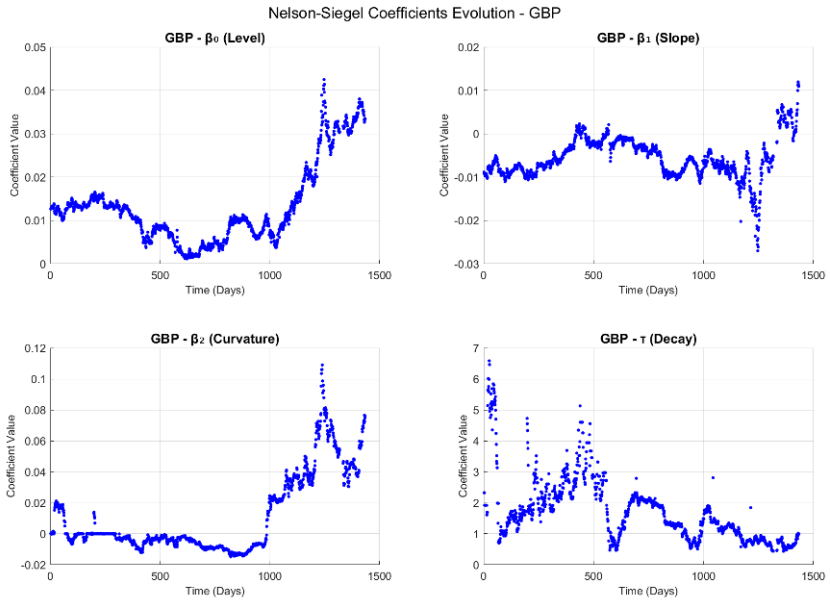


Figure A.8: Nelson-Siegel model coefficients – GBP

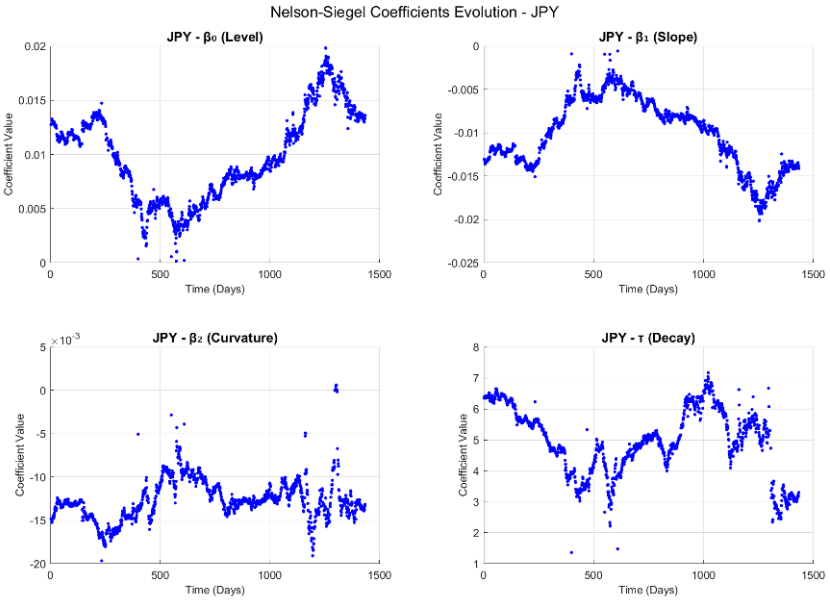


Figure A.9: Nelson-Siegel model coefficients – JPY

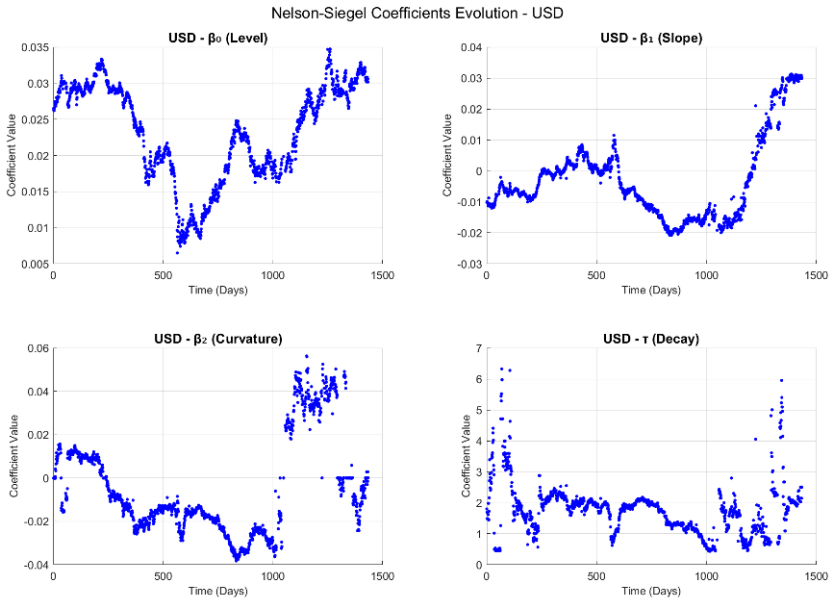


Figure A.10: Nelson-Siegel model coefficients – USD

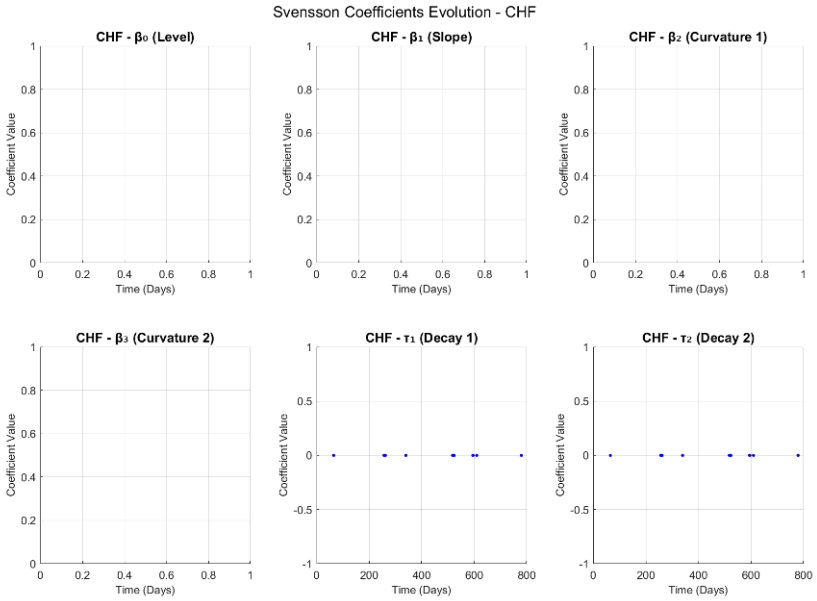


Figure A.11: Svensson model coefficients – CHF

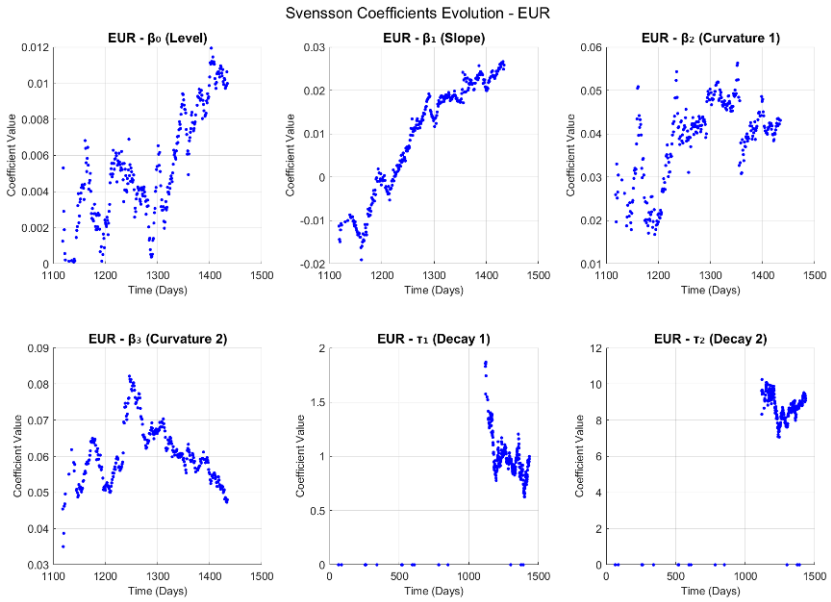


Figure A.12: Svensson model coefficients – EUR

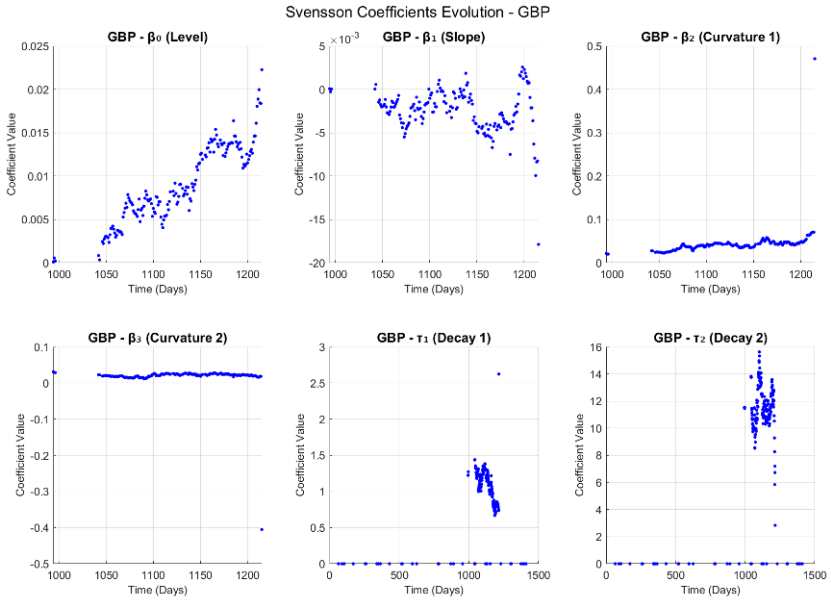


Figure A.13: Svensson model coefficients – GBP

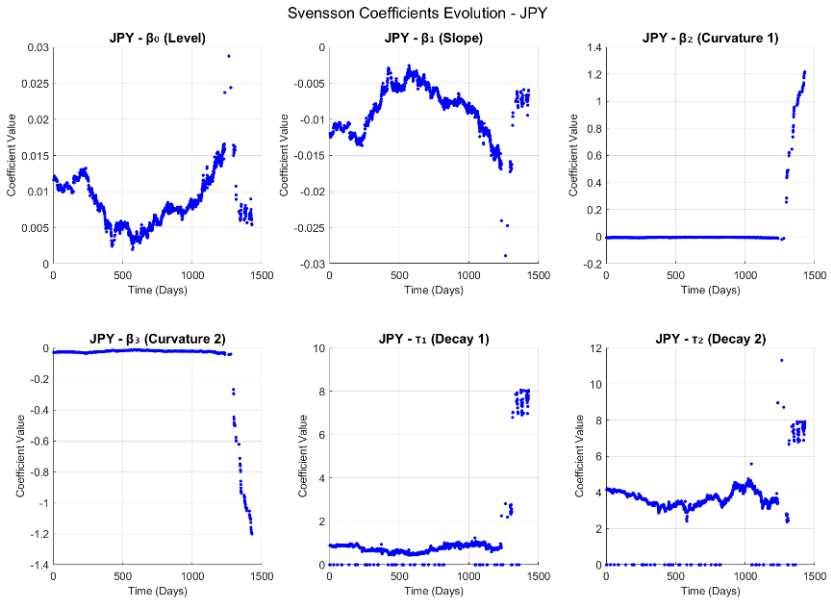


Figure A.14: Svensson model coefficients – JPY

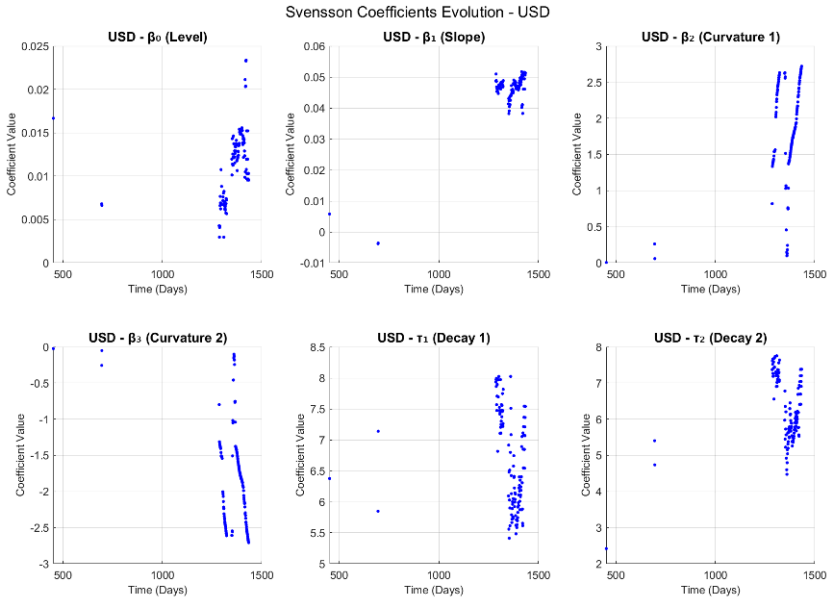


Figure A.15: Svensson model coefficients – USD

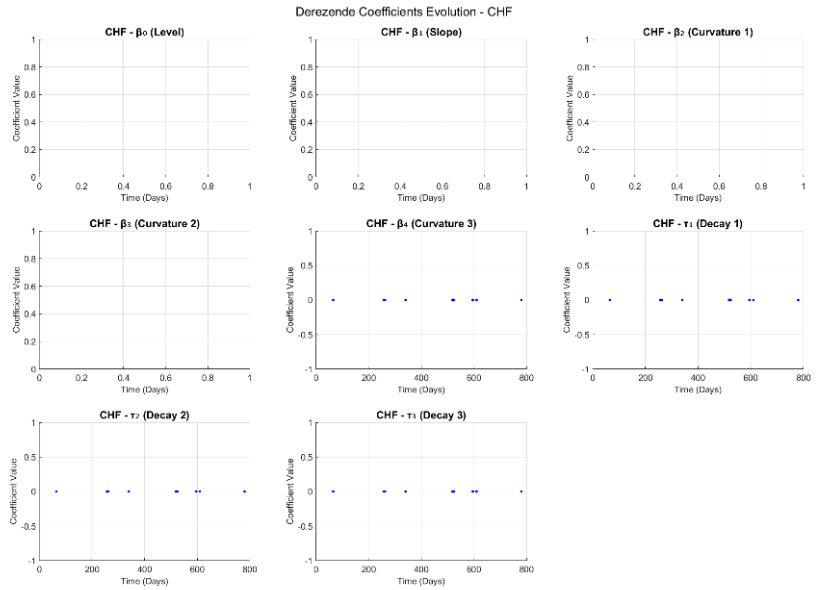


Figure A.16: De Rezende-Ferreira model coefficients – CHF

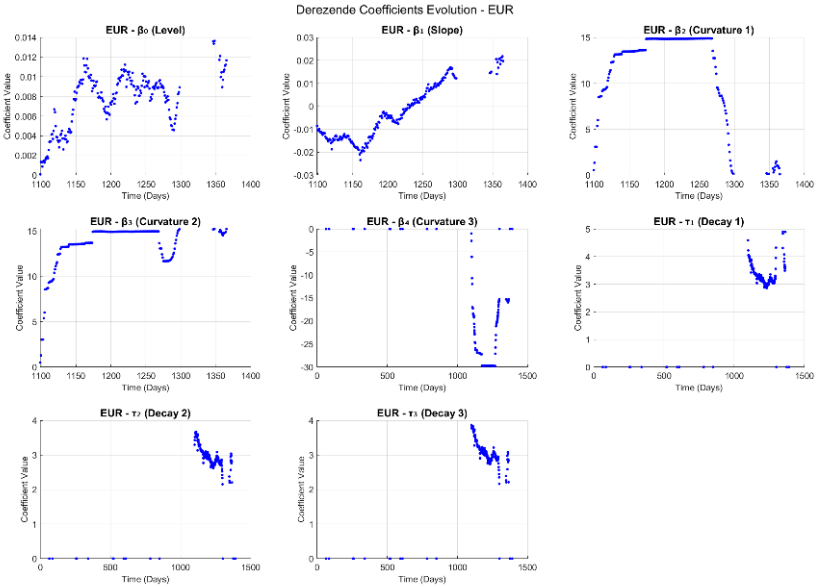


Figure A.17: De Rezende-Ferreira model coefficients – EUR

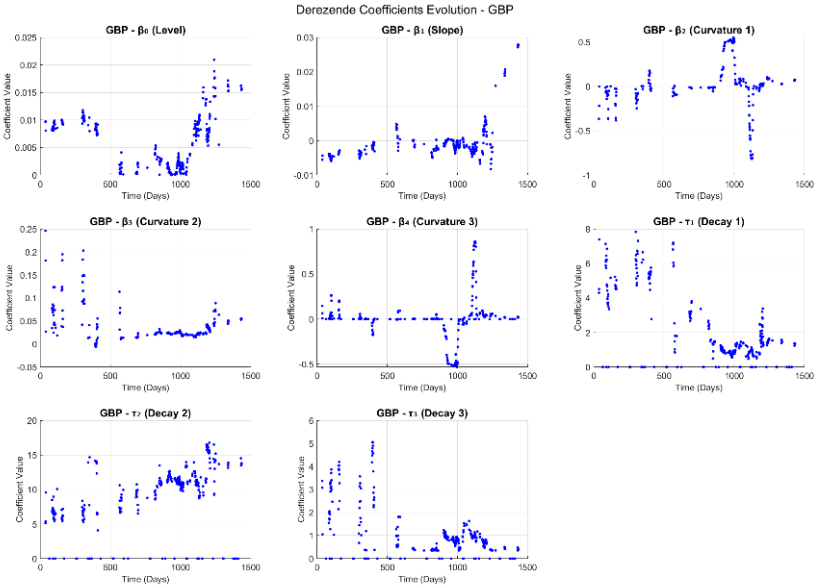


Figure A.18: De Rezende-Ferreira model coefficients – GBP

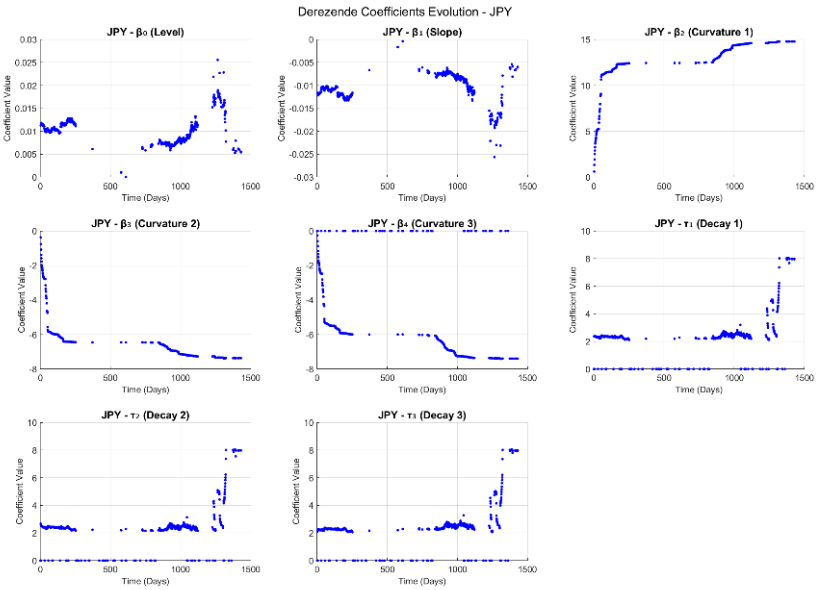


Figure A.19: De Rezende-Ferreira model coefficients – JPY

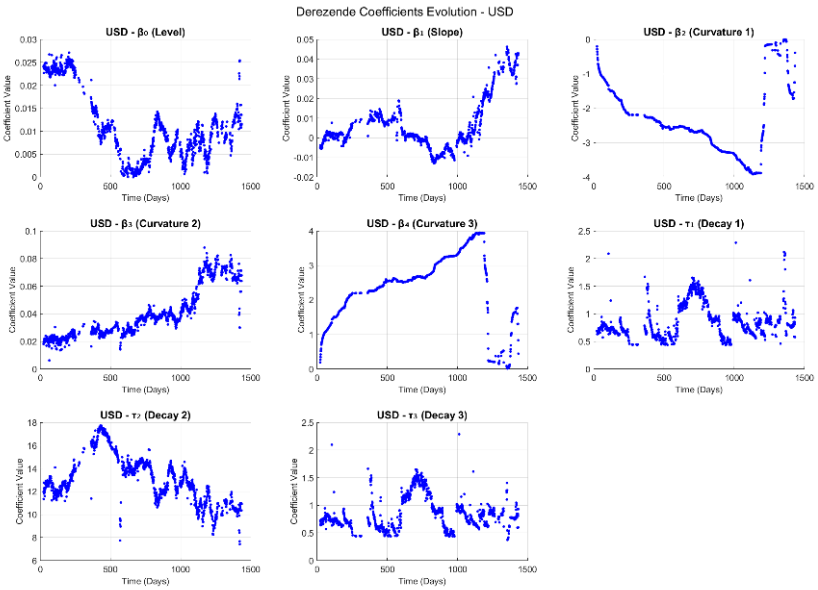


Figure A.20: De Rezende-Ferreira model coefficients – USD

A.1.3 Optimized Models (Developed Economies)

The following plots display in blue the estimated coefficients for each parameter, corresponding to the yield curves successfully fitted across the selected currencies. Coefficients shown in green refer to those curves that were successfully fitted with the assistance of the genetic algorithm solver, while those in red were obtained through optimization using particle swarm methods.

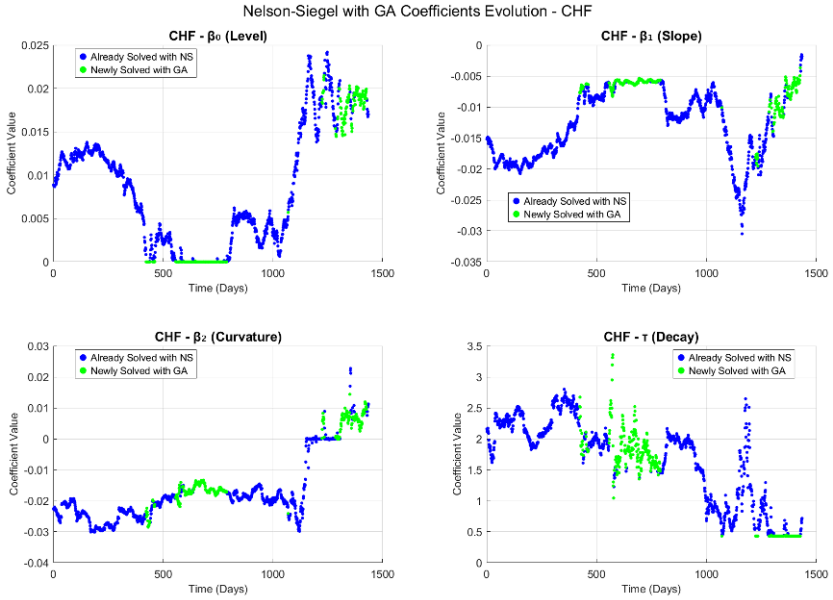


Figure A.21: Nelson-Siegel model (GA-optimized) coefficients – CHF

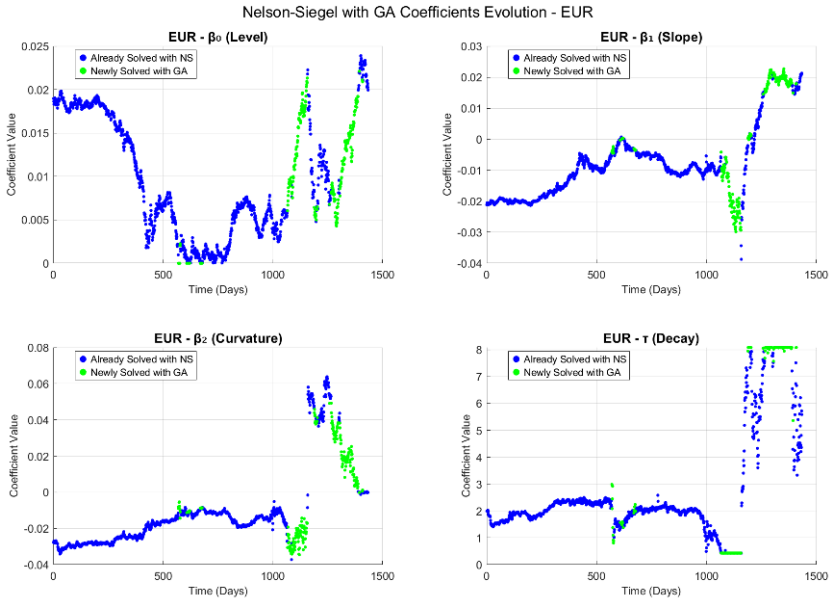


Figure A.22: Nelson-Siegel model (GA-optimized) coefficients – EUR

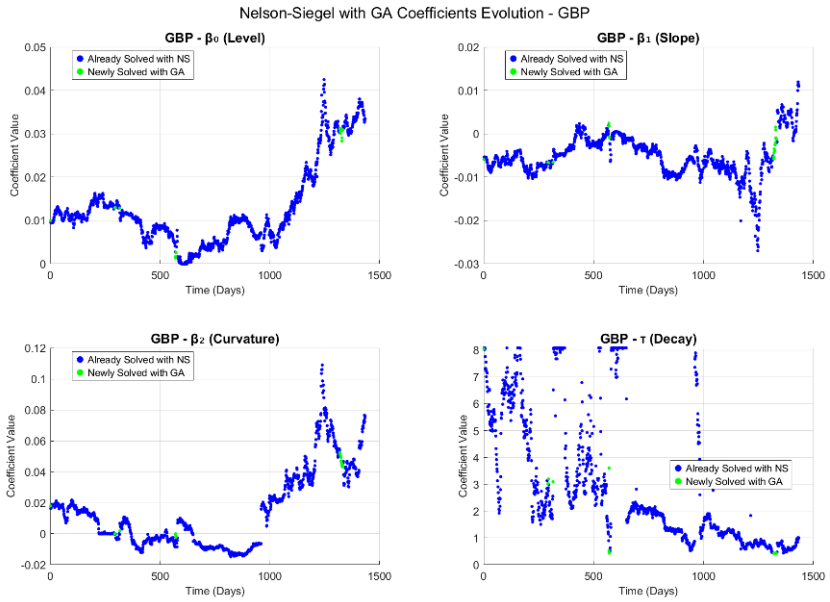


Figure A.23: Nelson-Siegel model (GA-optimized) coefficients – GBP

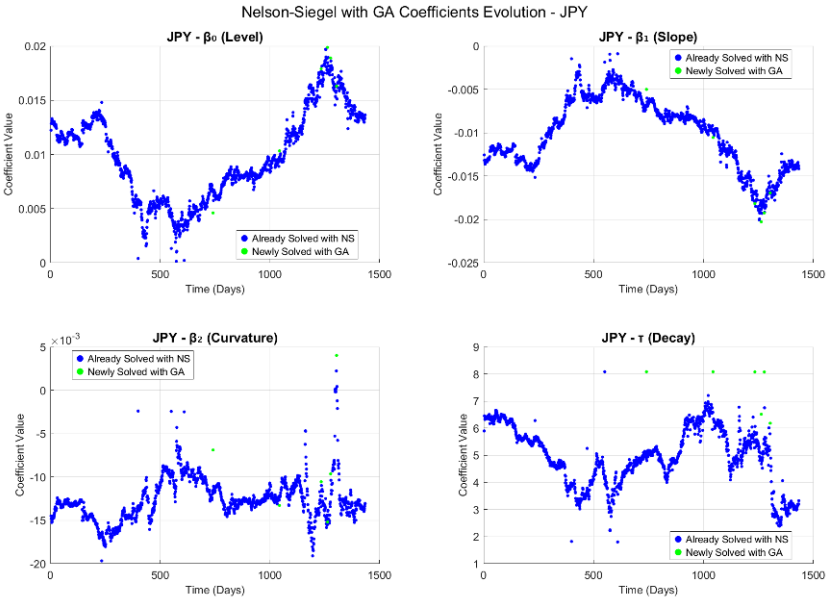


Figure A.24: Nelson-Siegel model (GA-optimized) coefficients – JPY

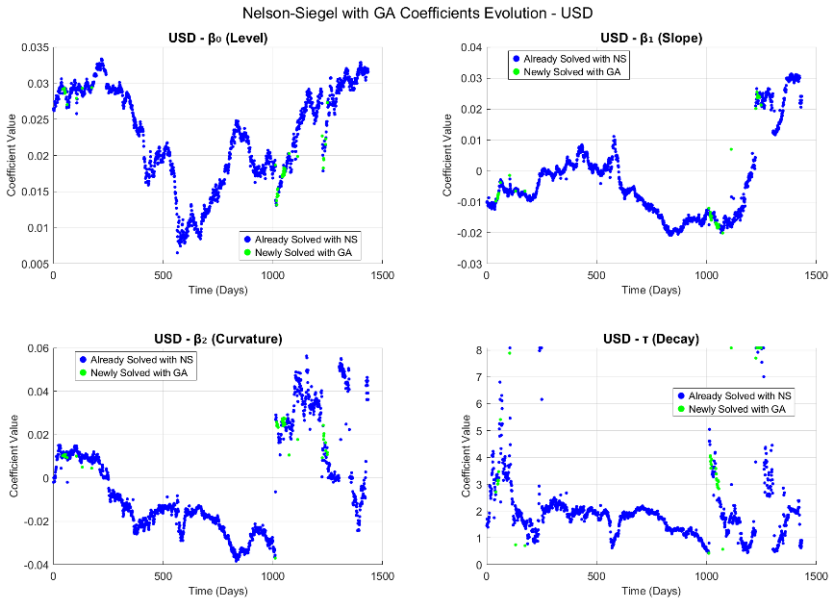


Figure A.25: Nelson-Siegel model (GA-optimized) coefficients – USD

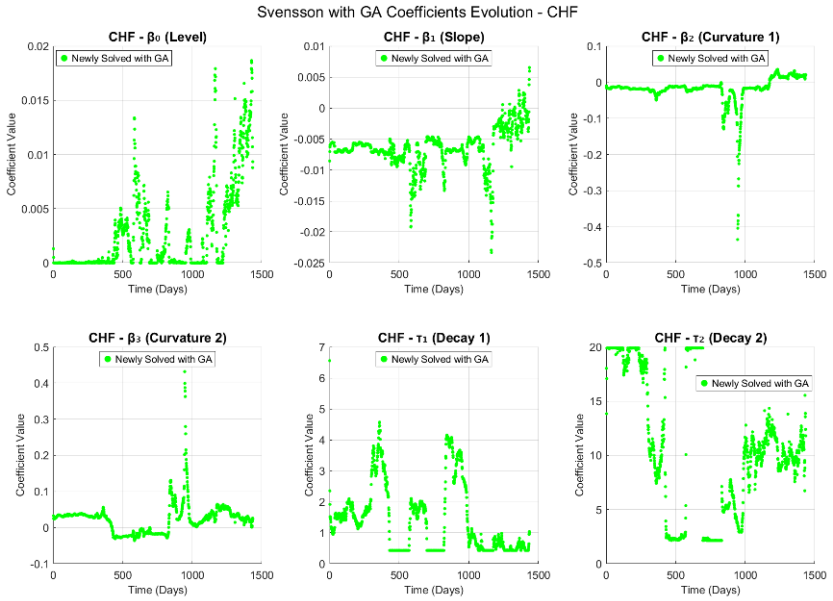


Figure A.26: Svensson model (GA-optimized) coefficients – CHF

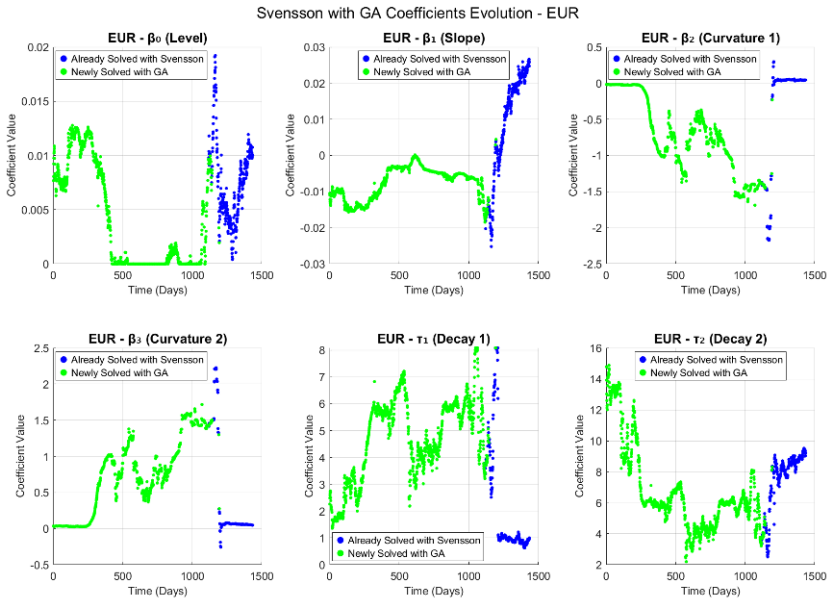


Figure A.27: Svensson model (GA-optimized) coefficients – EUR

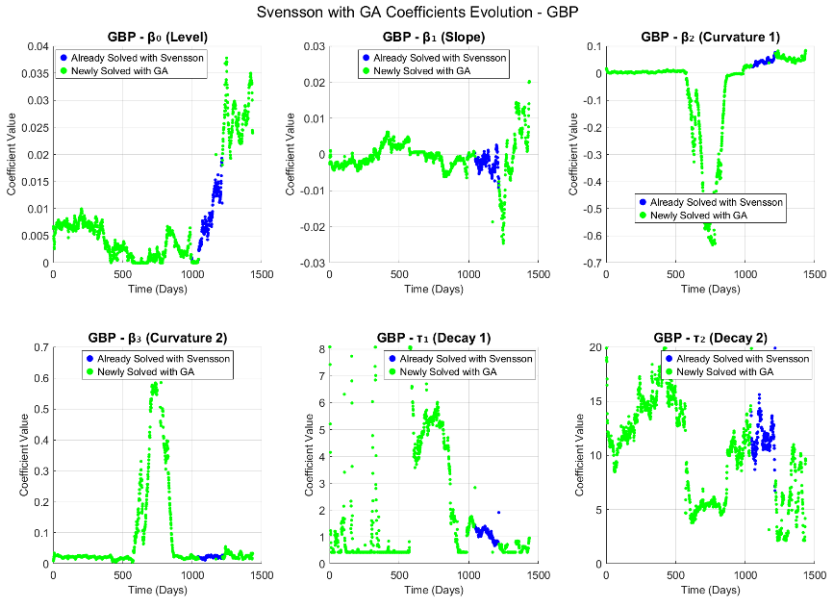


Figure A.28: Svensson model (GA-optimized) coefficients – GBP

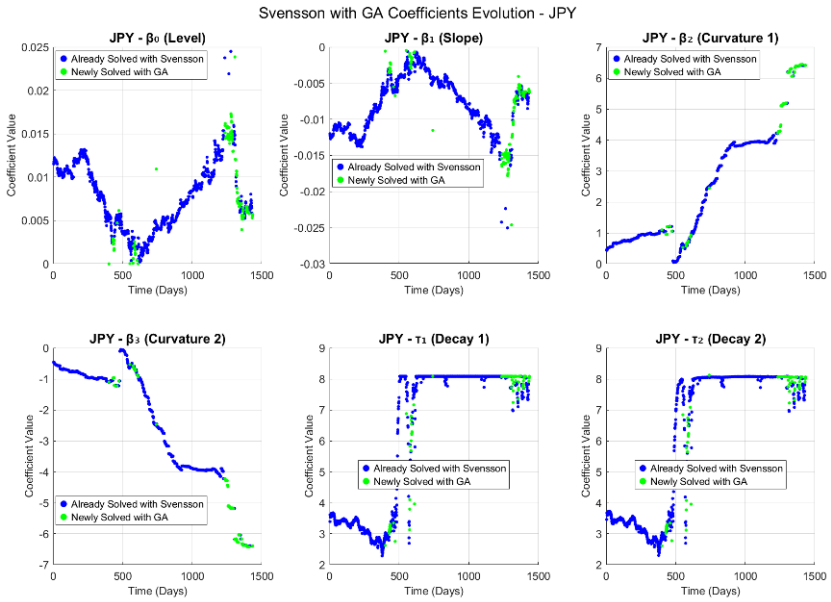


Figure A.29: Svensson model (GA-optimized) coefficients – JPY

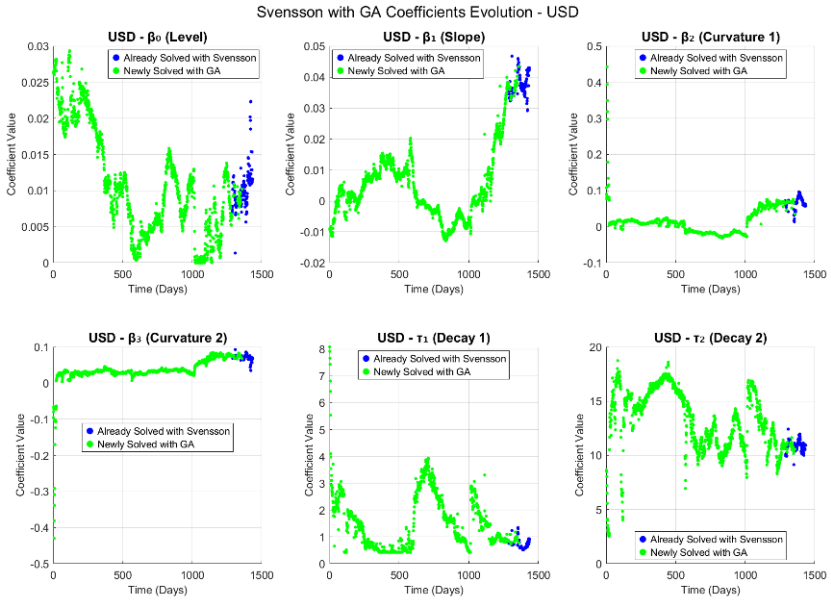


Figure A.30: Svensson model (GA-optimized) coefficients – USD

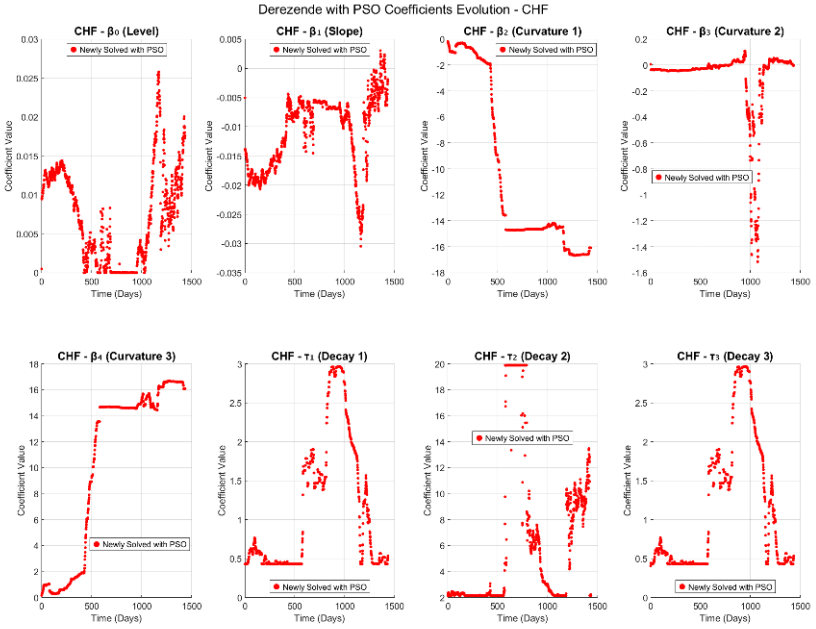


Figure A.31: De Rezende-Ferreira model (PSO-optimized) coefficients – CHF

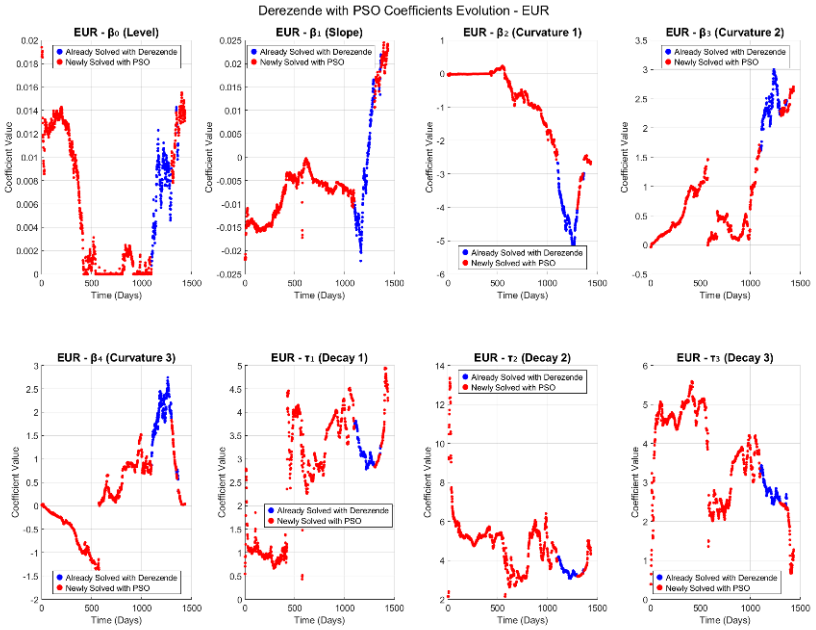


Figure A.32: De Rezende-Ferreira model (PSO-optimized) coefficients – EUR

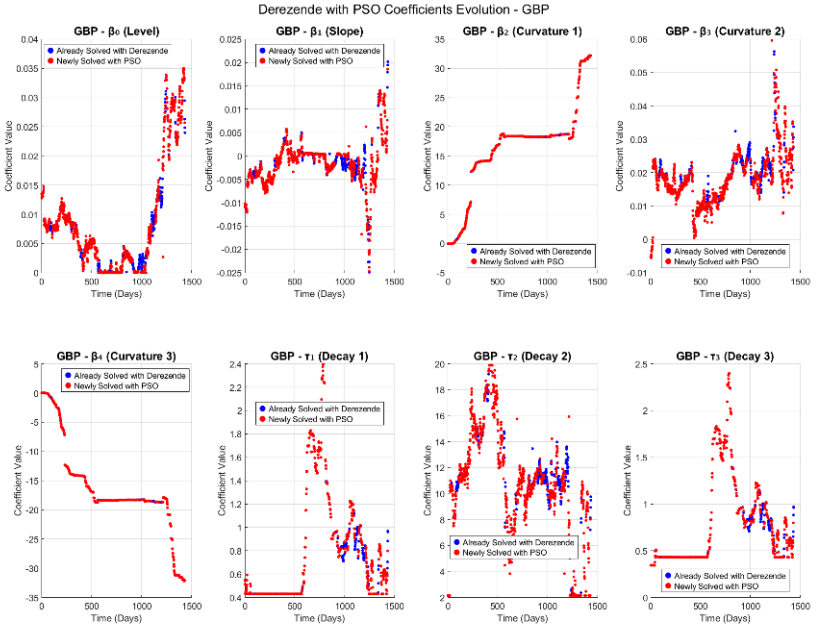


Figure A.33: De Rezende-Ferreira model (PSO-optimized) coefficients – GBP

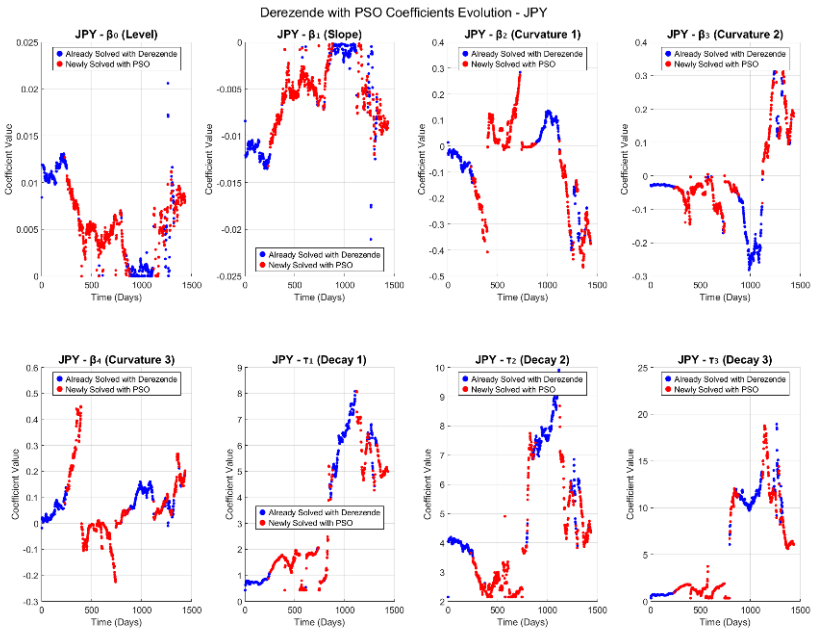


Figure A.34: De Rezende-Ferreira model (PSO-optimized) coefficients – JPY

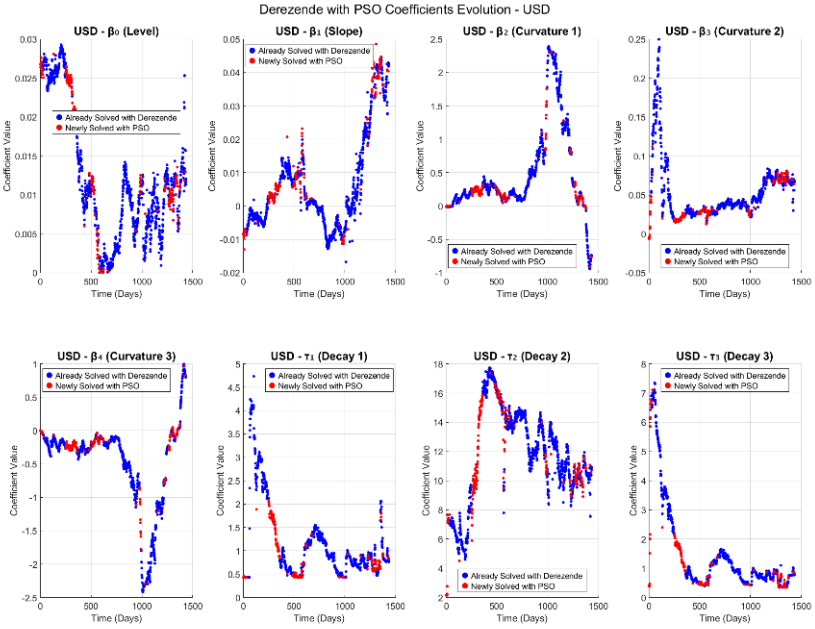


Figure A.35: De Rezende-Ferreira model (PSO-optimized) coefficients – USD

A.1.4 GPR and further Results (Developed Economies)

Below we report the in-sample and out-of-sample RMSE time series obtained by fitting the yield curves using the Gaussian Process Regression (GPR) model.

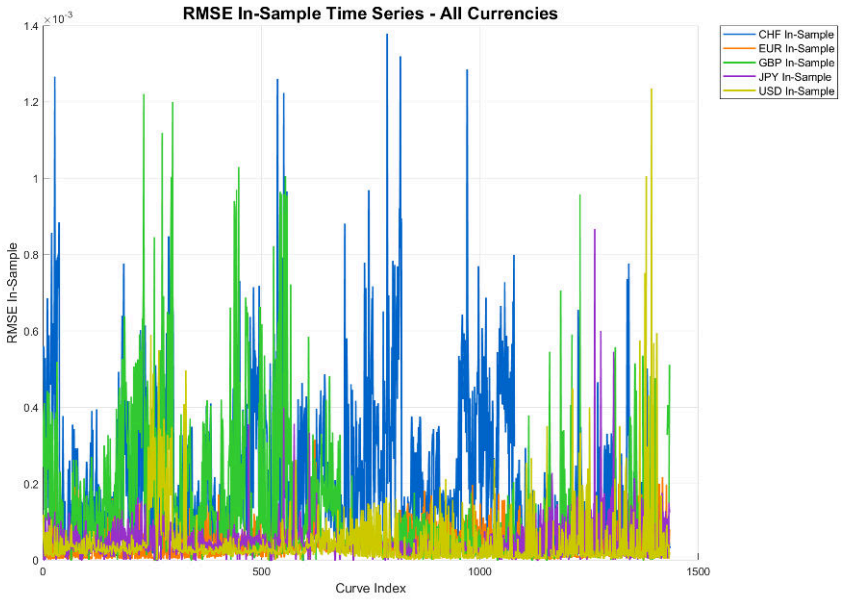


Figure A.36: In-sample RMSE time series – GPR model (all currencies)

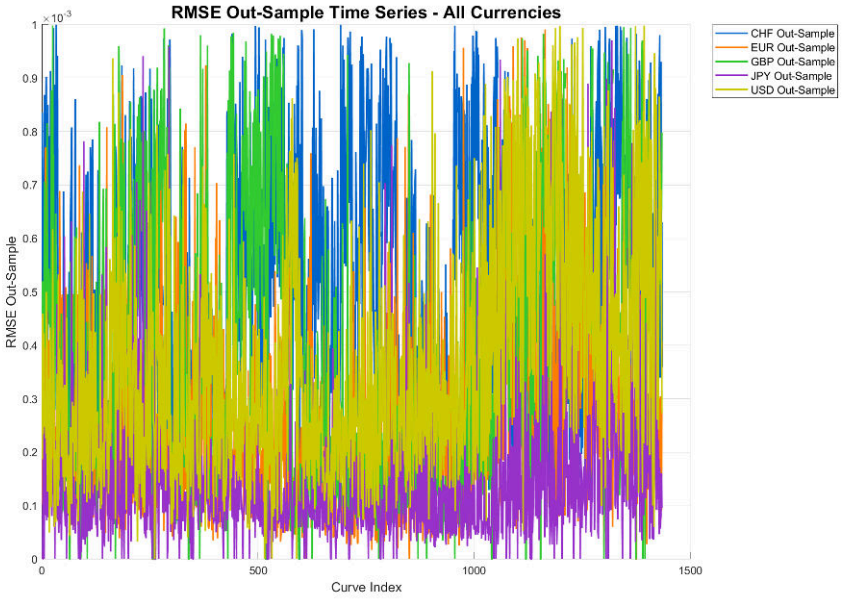


Figure A.37: Out-of-sample RMSE time series – GPR model (all currencies)

To conclude, the following histograms summarize the percentage of

yield curves successfully fitted using each method, both across individual currencies and in aggregate.

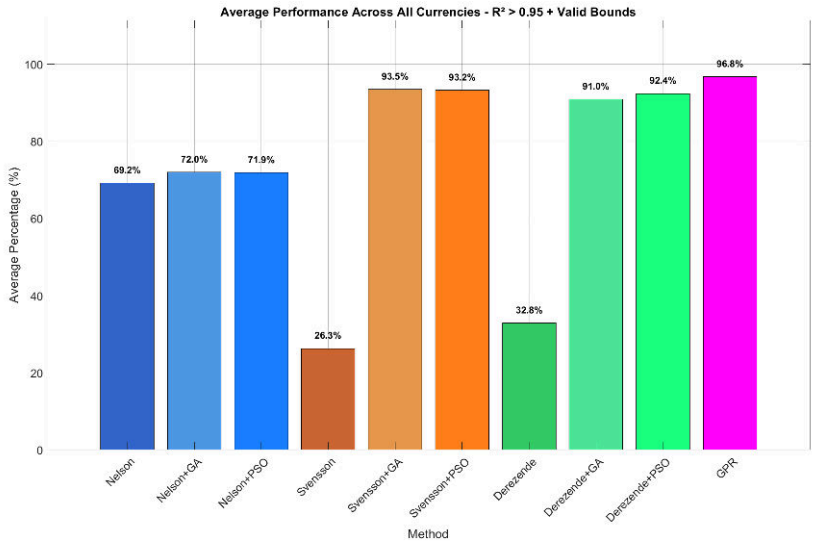


Figure A.38: Average performance of all methods

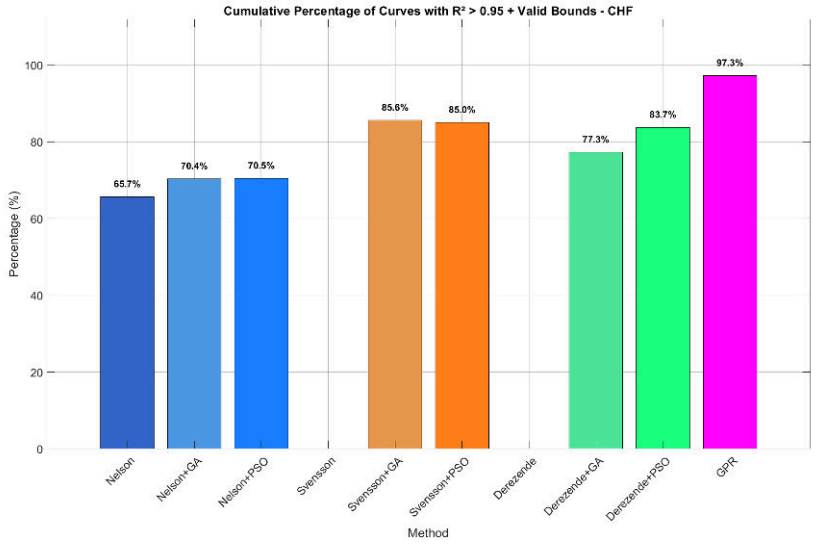


Figure A.39: Cumulative R^2 bounds – CHF

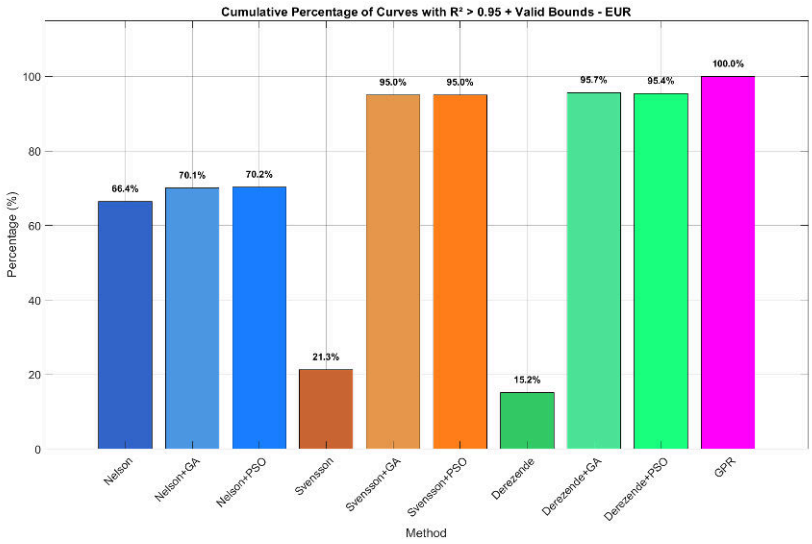


Figure A.40: Cumulative R^2 bounds – EUR

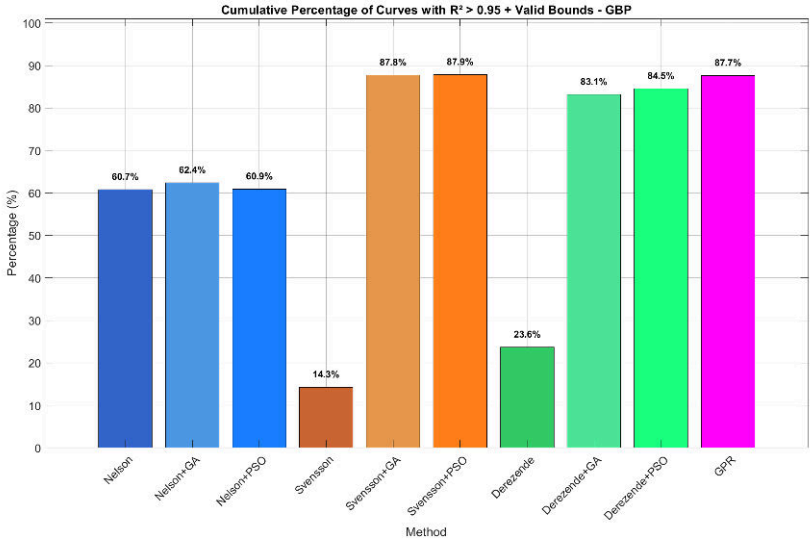


Figure A.41: Cumulative R^2 bounds – GBP

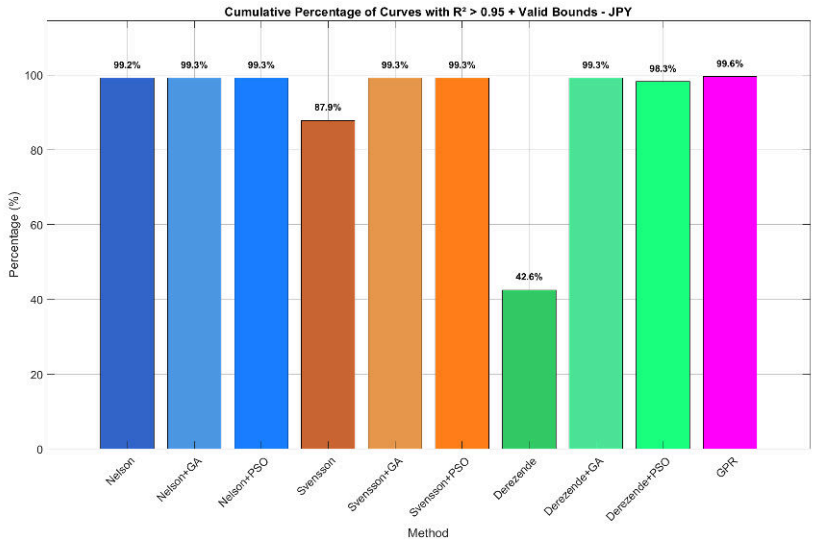


Figure A.42: Cumulative R^2 bounds – JPY

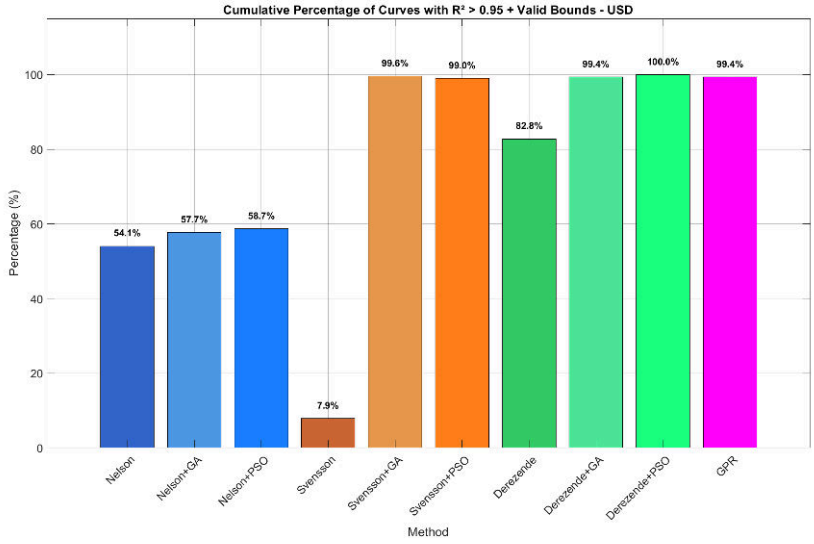


Figure A.43: Cumulative R^2 bounds – USD

A.2 Main Pacific Economies

The following figures present the yield curve surfaces for the currencies associated with the main Pacific economies.

A.2.1 IR term structures (Pacific Economies)

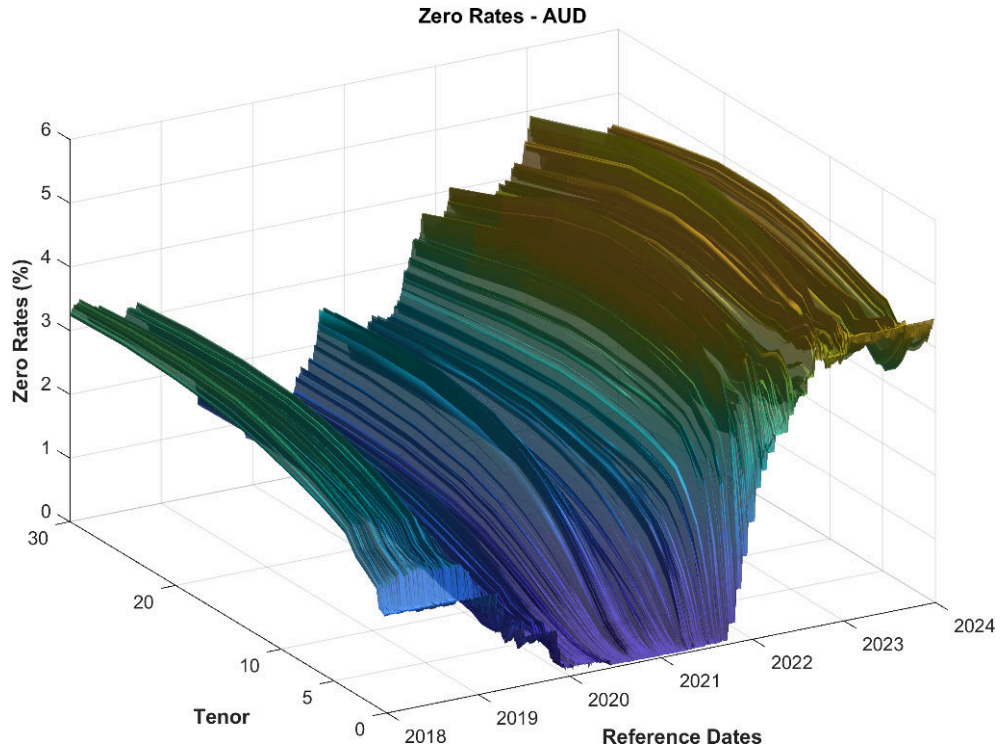


Figure A.44: Interest rates term structure – AUD

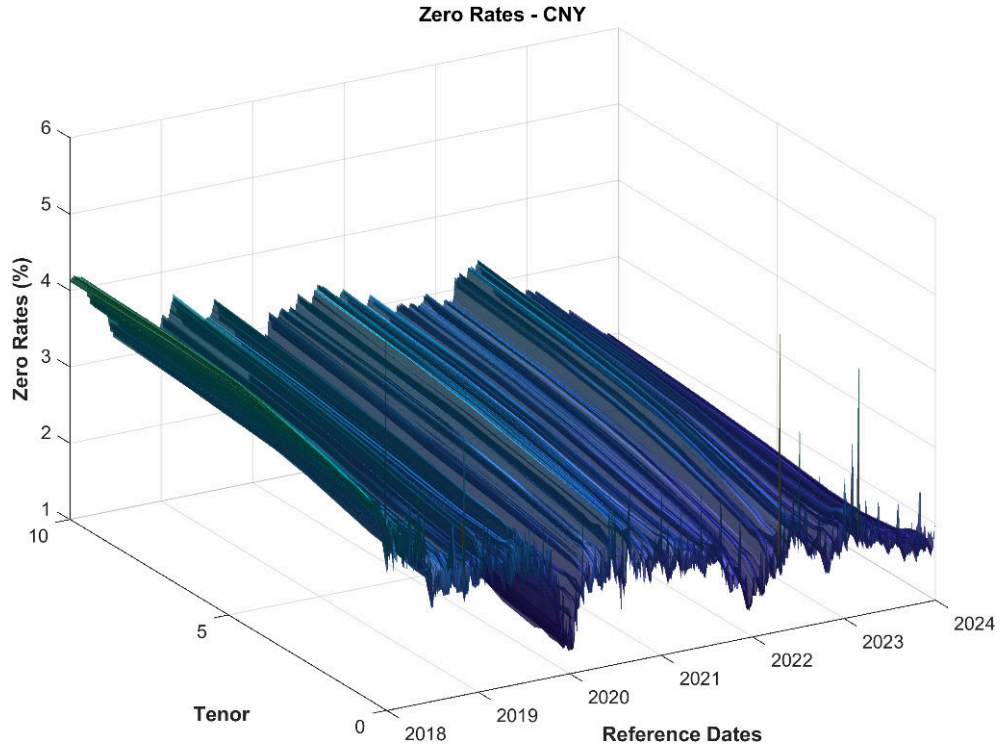


Figure A.45: Interest rates term structure – CNY

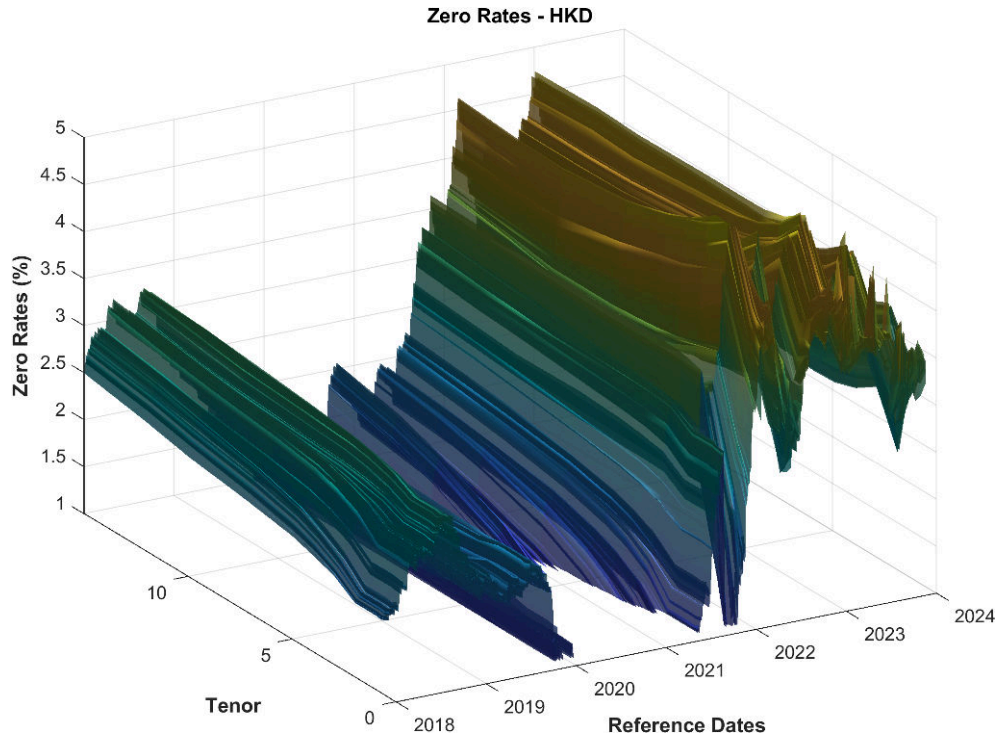


Figure A.46: Interest rates term structure – HKD

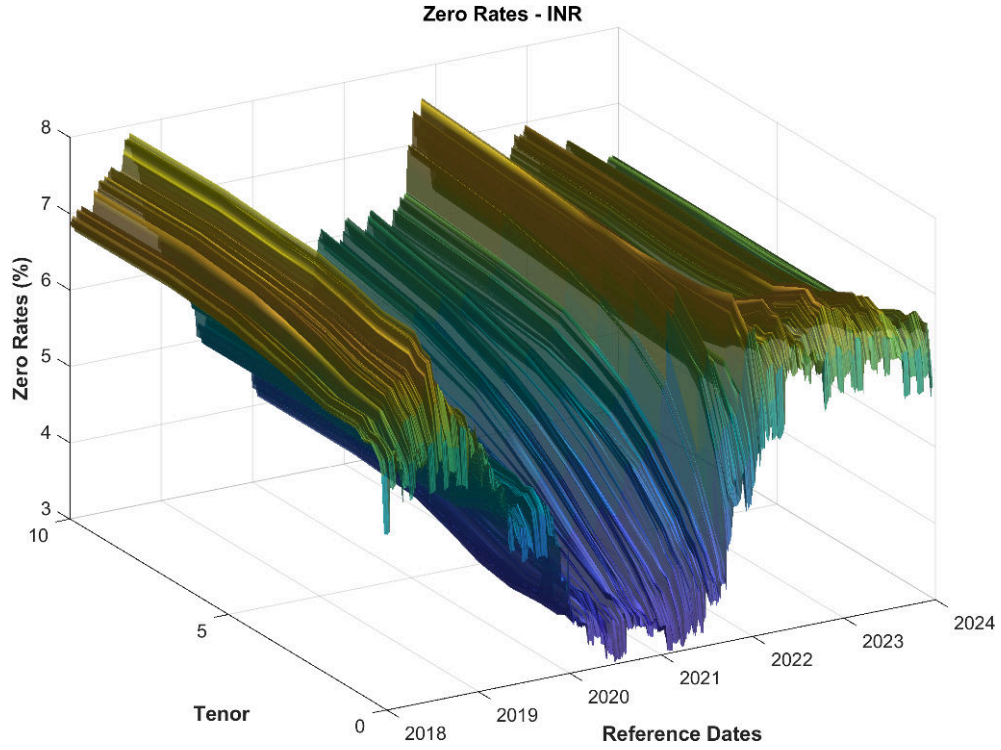


Figure A.47: Interest rates term structure – INR

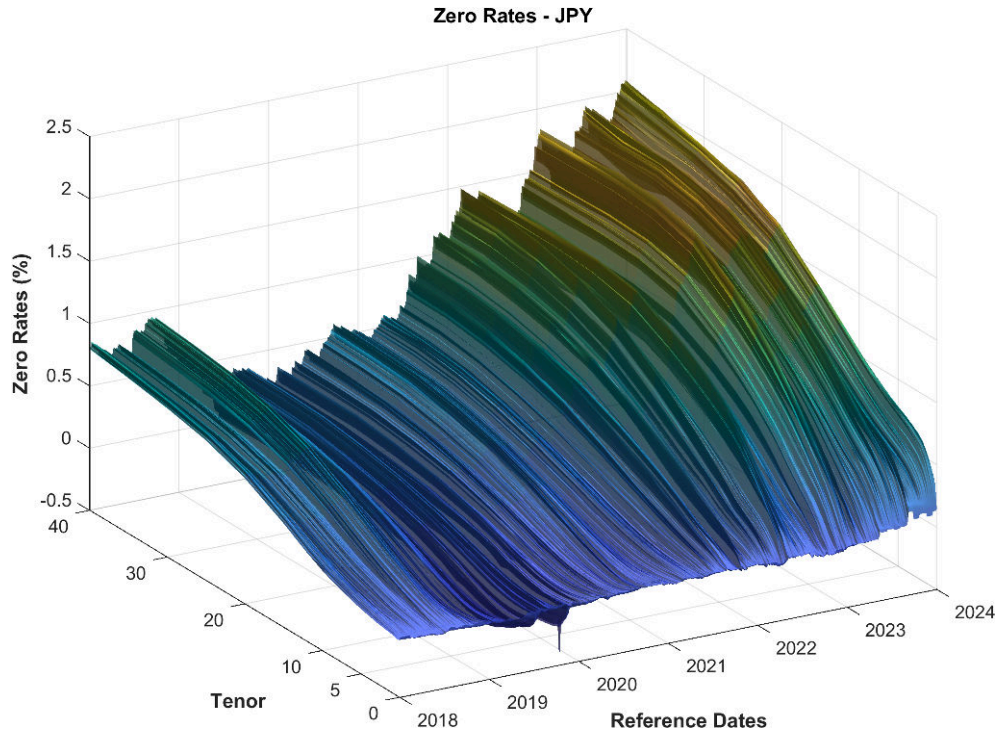


Figure A.48: Interest rates term structure – JPY

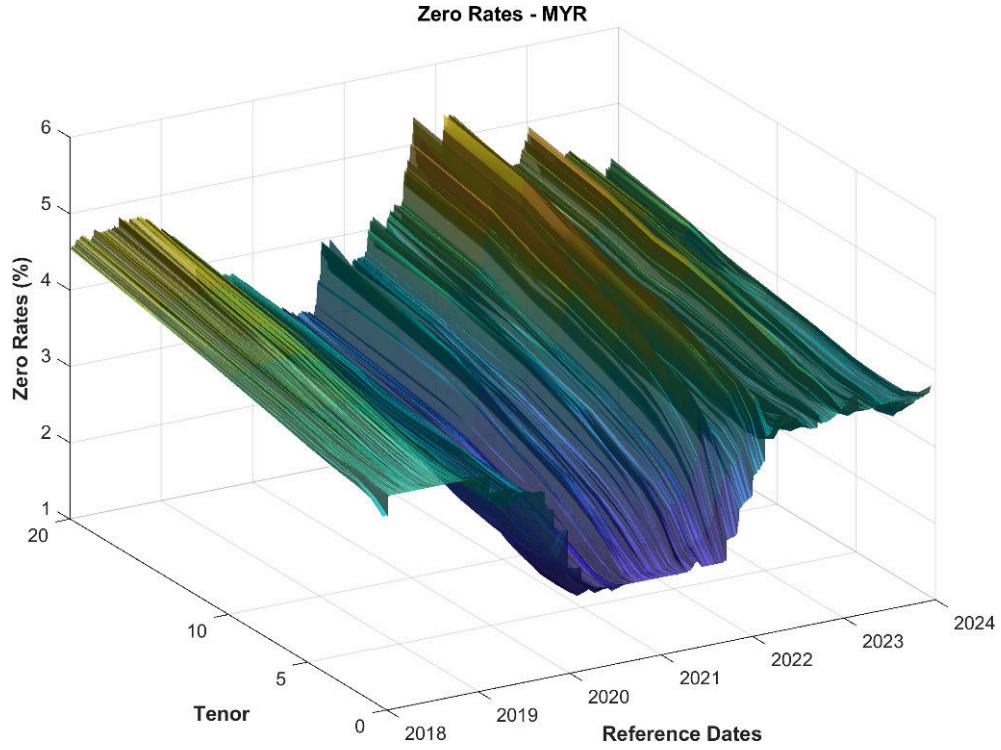


Figure A.49: Interest rates term structure – MYR

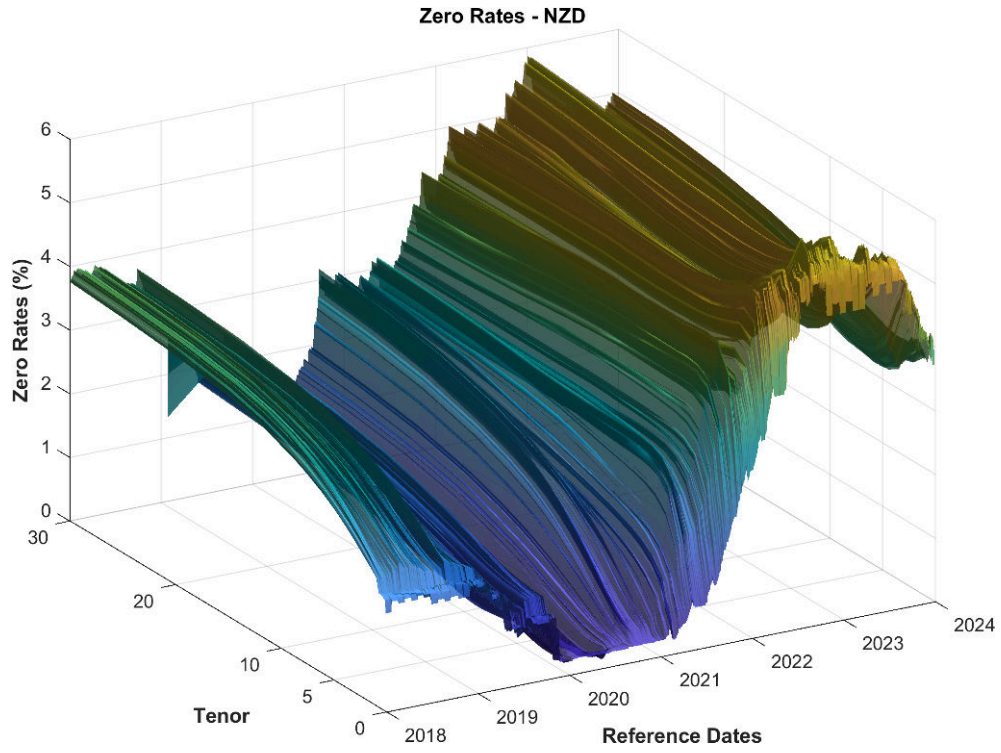


Figure A.50: Interest rates term structure – NZD

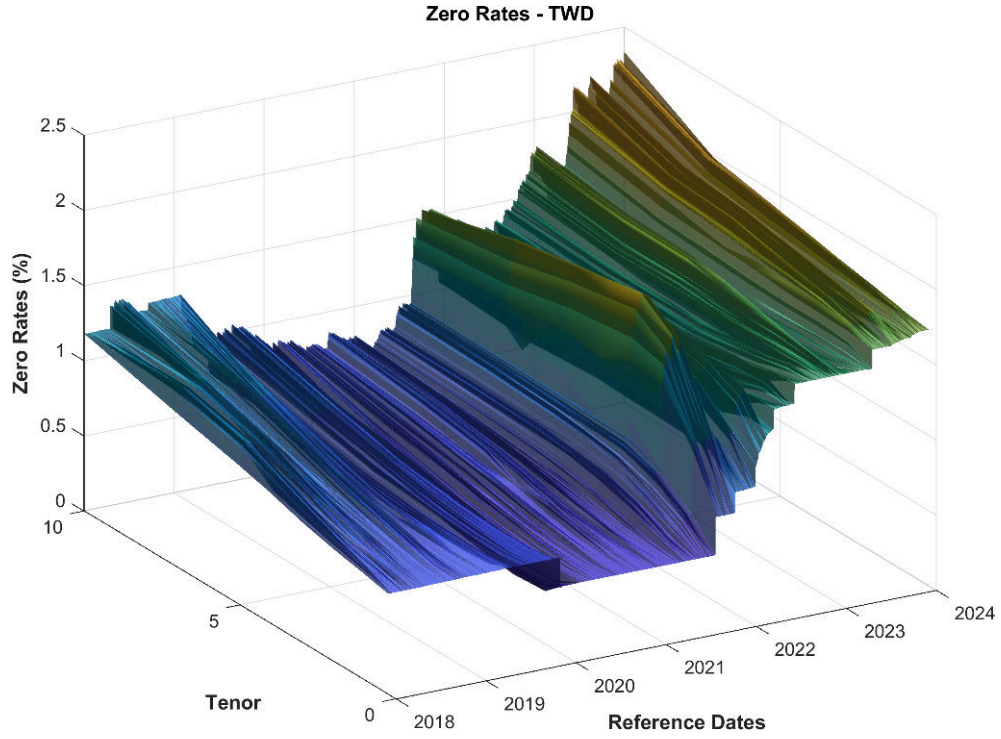


Figure A.51: Interest rates term structure – TWD

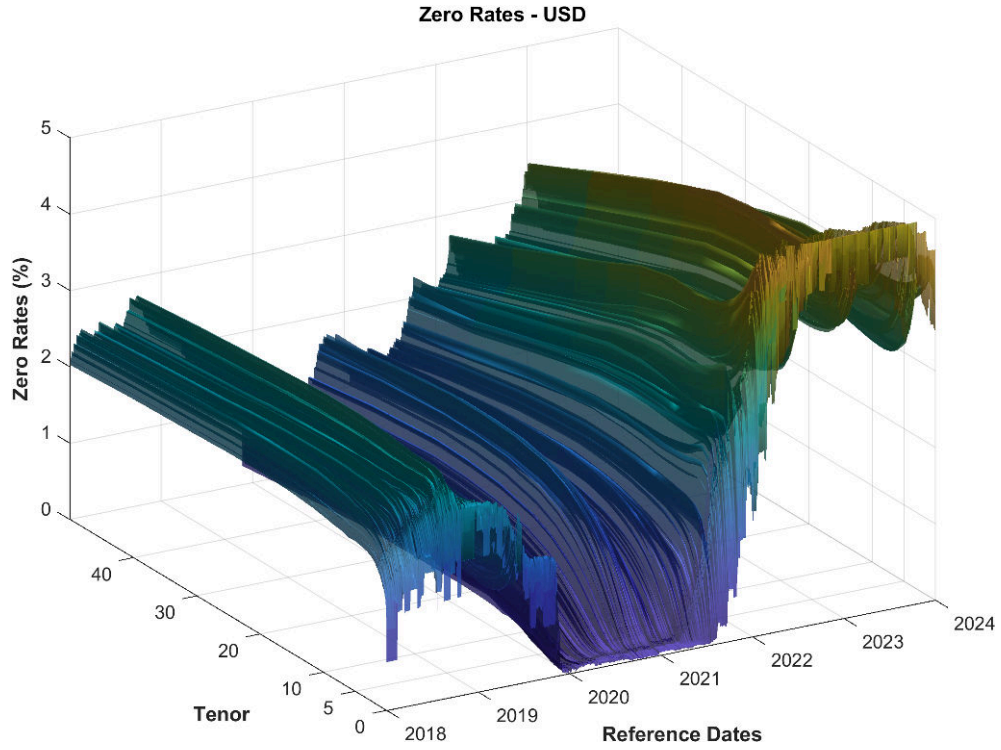


Figure A.52: Interest rates term structure – USD

A.2.2 Base Models (Pacific Economies)

The following plots display the estimated coefficients for each parameter involved in the curve fitting process. Coefficients shown in blue correspond to curves successfully fitted using standard optimization. Those in green were obtained with the support of a genetic algorithm (GA), while red values reflect results achieved via particle swarm optimization (PSO). Coefficients in orange indicate curves fitted through simulated annealing (SA)

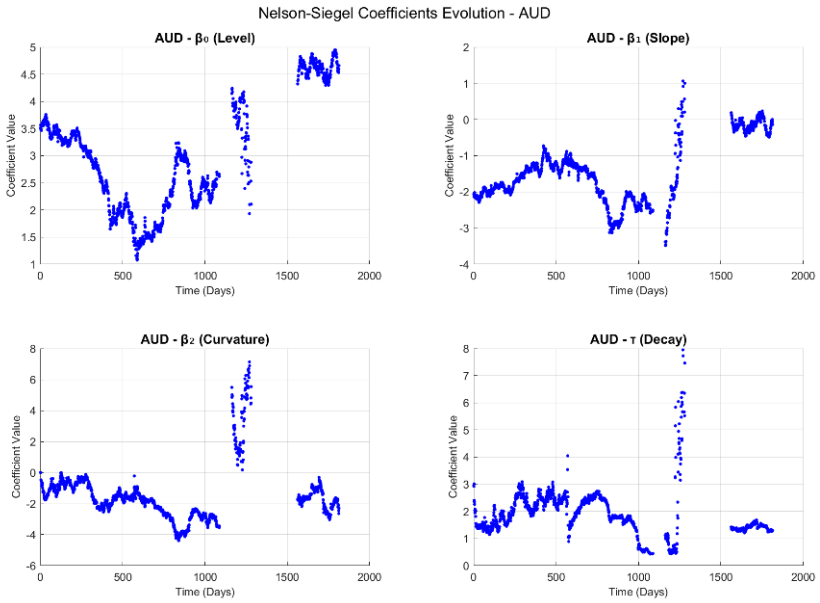


Figure A.53: Nelson model coefficients – AUD

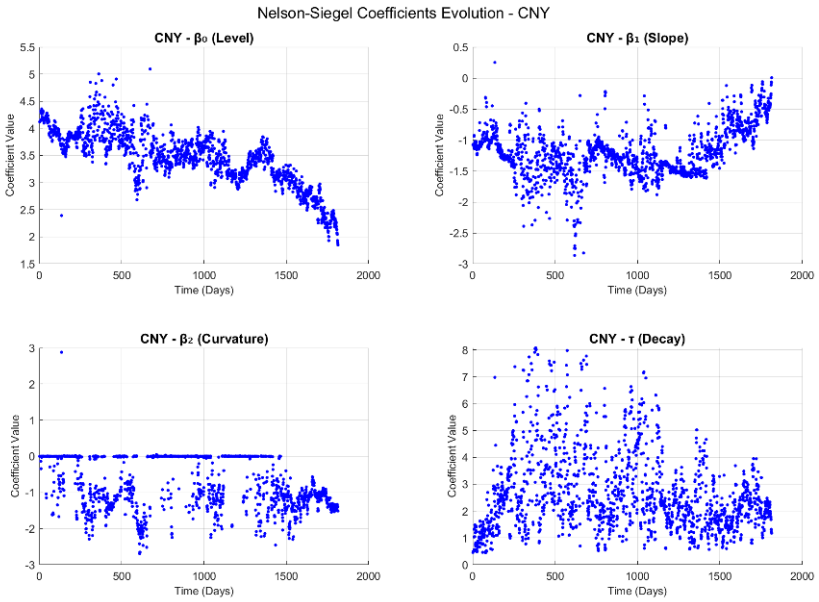


Figure A.54: Nelson model coefficients – CNY

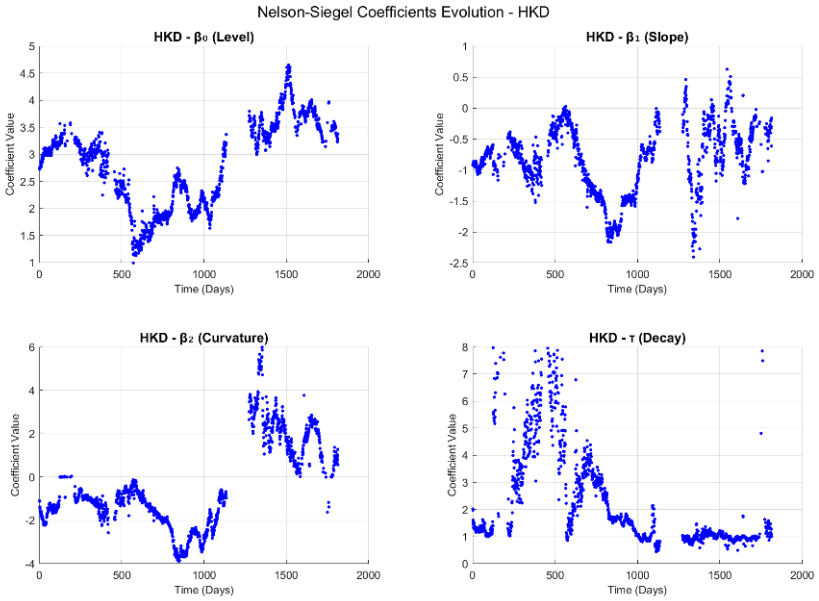


Figure A.55: Nelson model coefficients – HKD

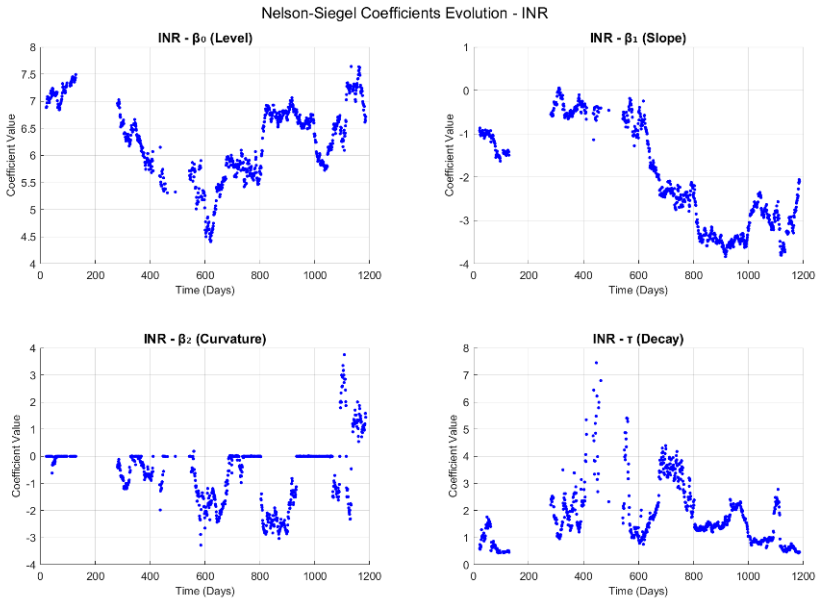


Figure A.56: Nelson model coefficients – INR

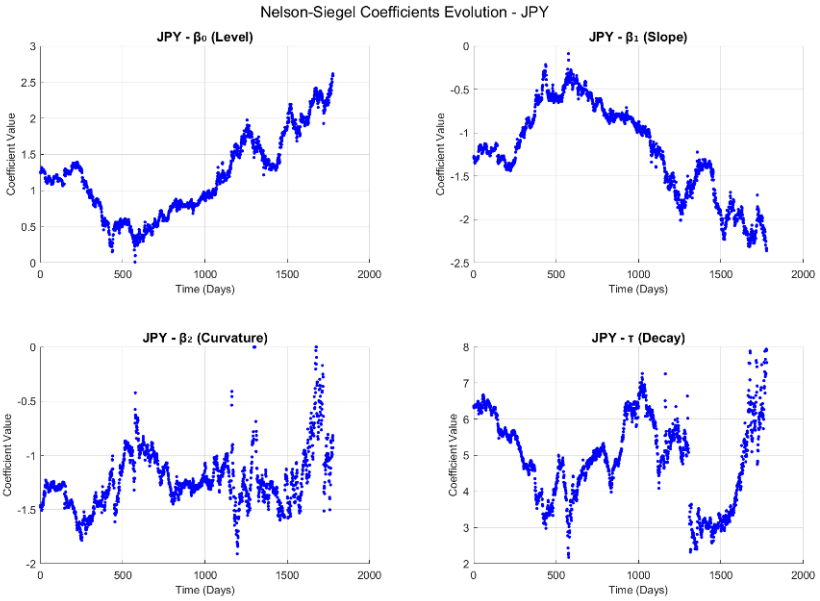


Figure A.57: Nelson model coefficients – JPY

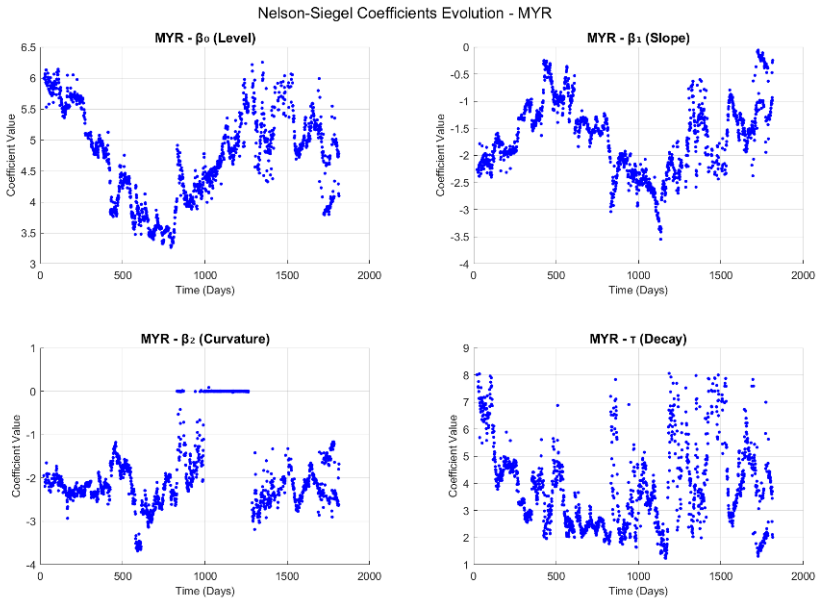


Figure A.58: Nelson model coefficients – MYR

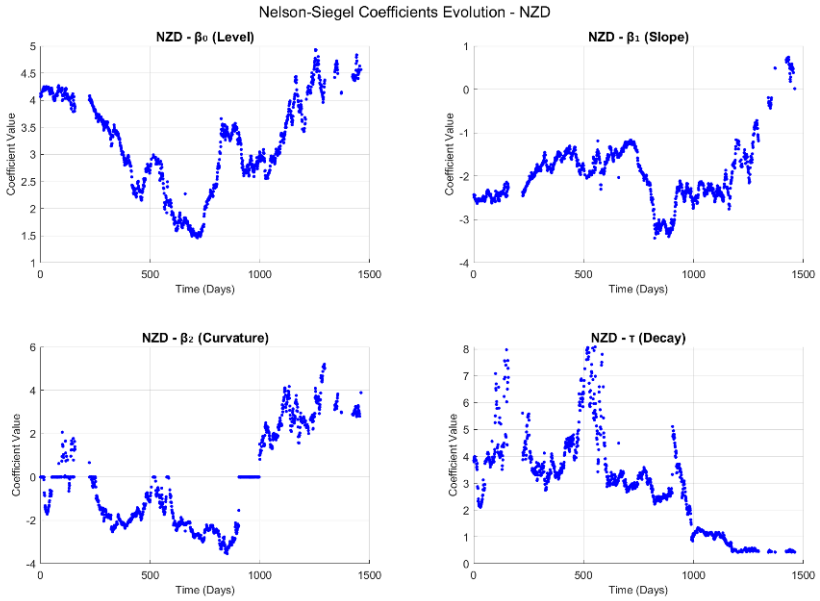


Figure A.59: Nelson model coefficients – NZD

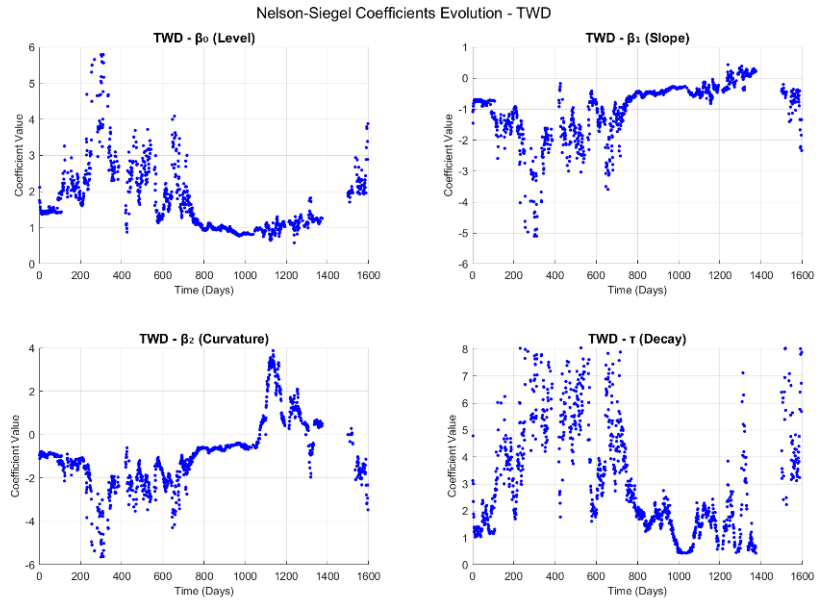


Figure A.60: Nelson model coefficients – TWD

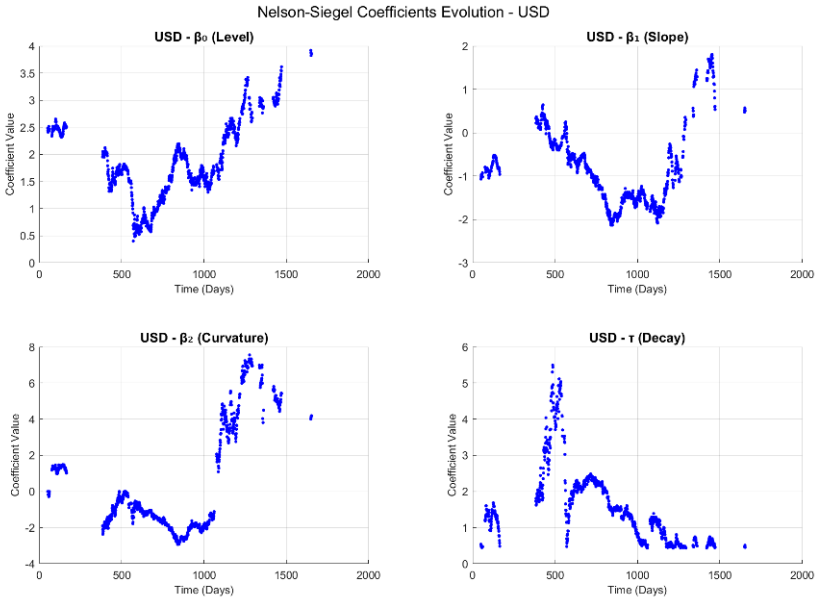


Figure A.61: Nelson model coefficients – USD

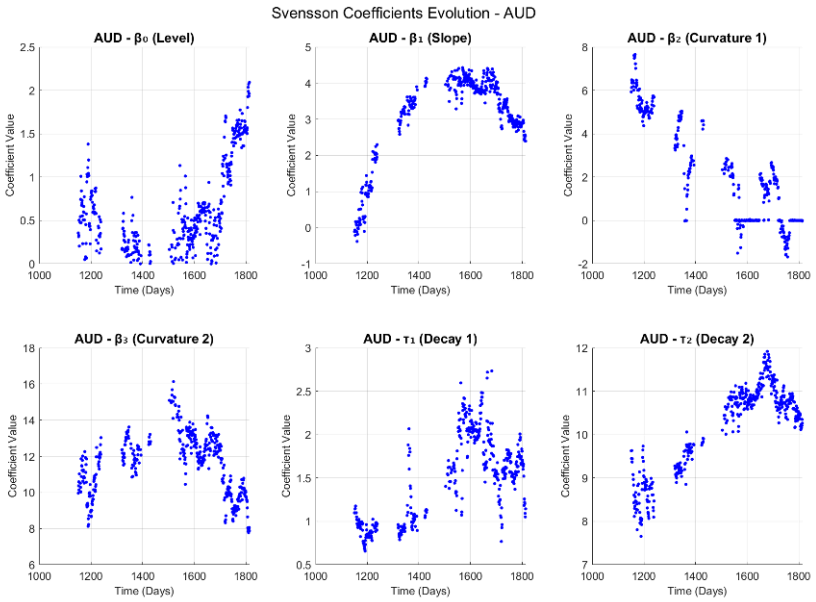


Figure A.62: Svensson model coefficients – AUD

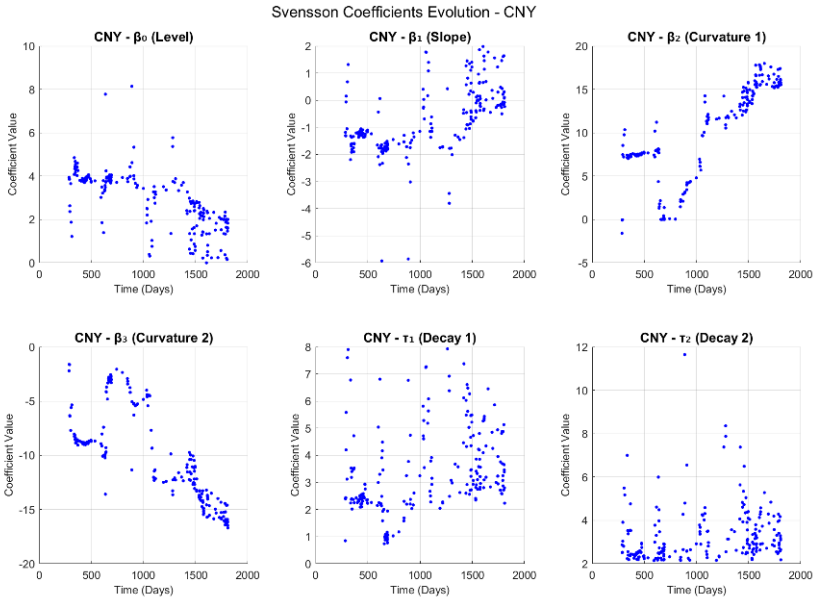


Figure A.63: Svensson model coefficients – CNY

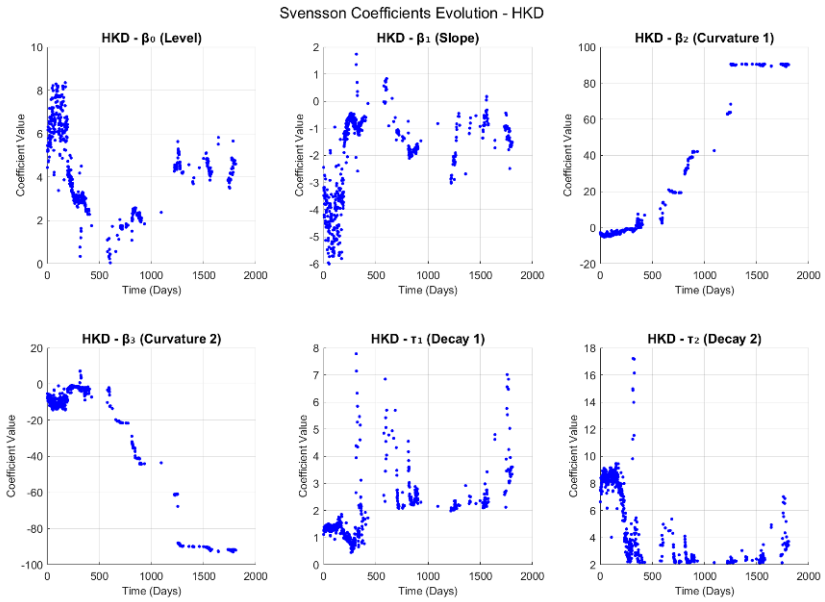


Figure A.64: Svensson model coefficients – HKD

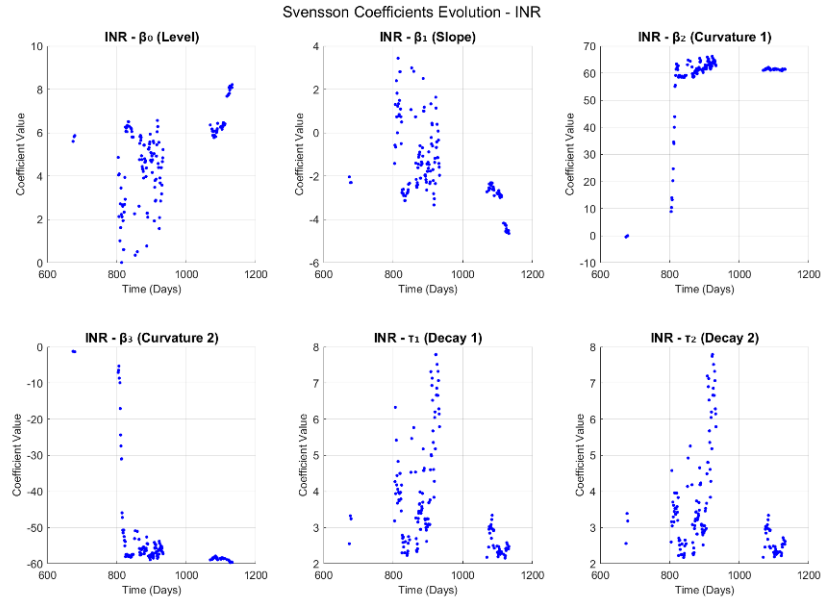


Figure A.65: Svensson model coefficients – INR

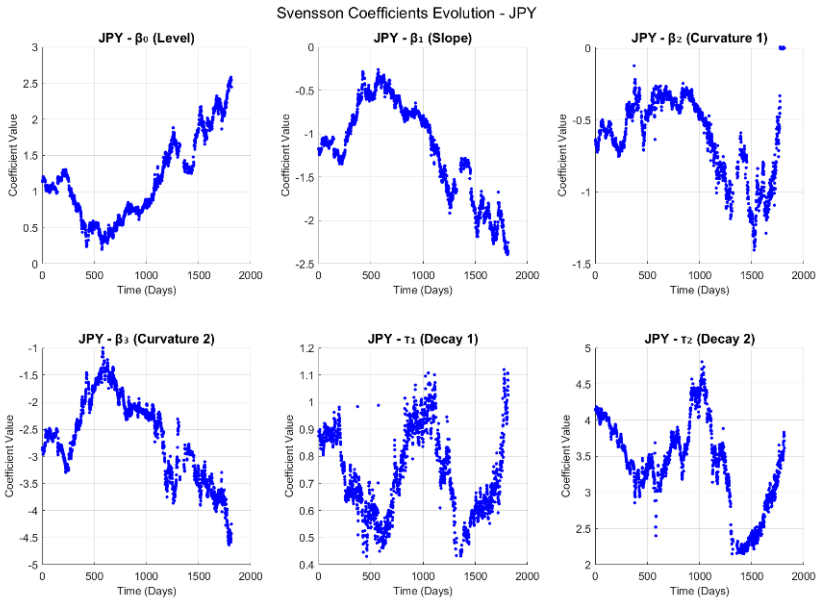


Figure A.66: Svensson model coefficients – JPY

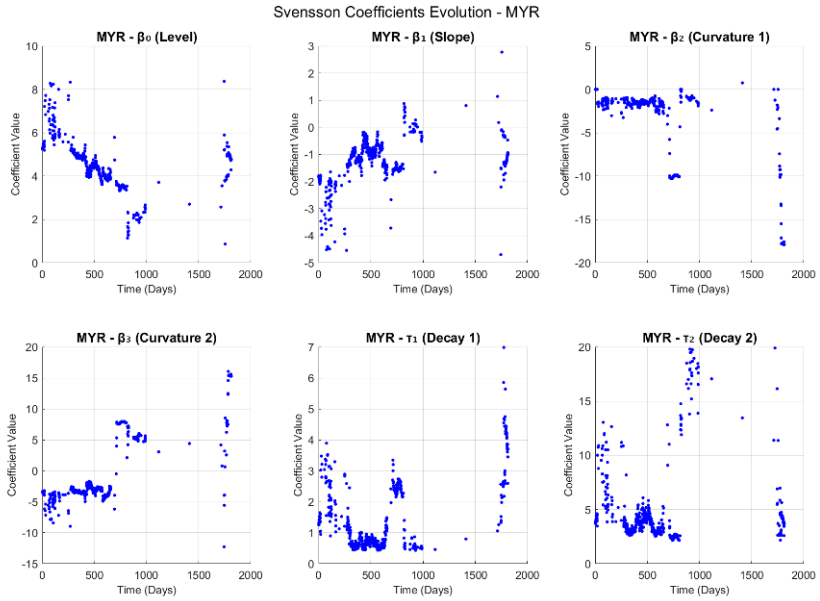


Figure A.67: Svensson model coefficients – MYR

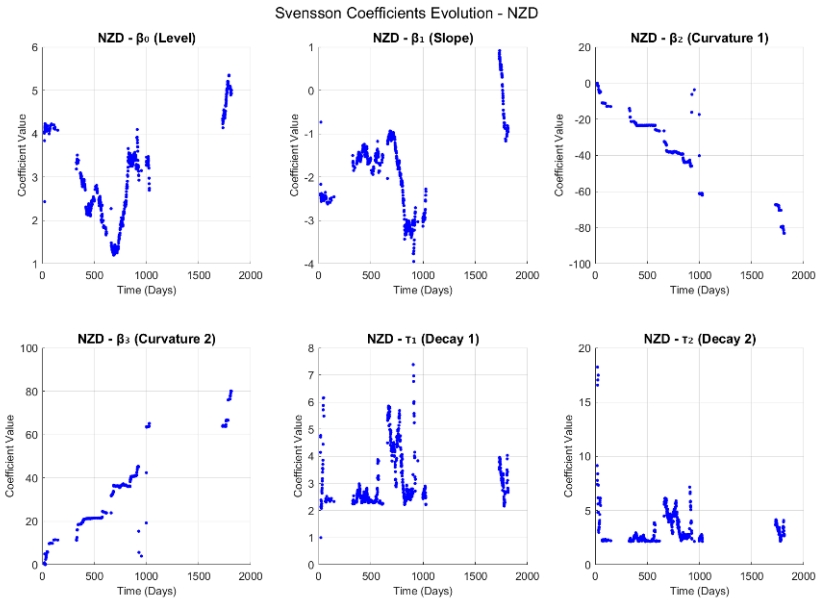


Figure A.68: Svensson model coefficients – NZD

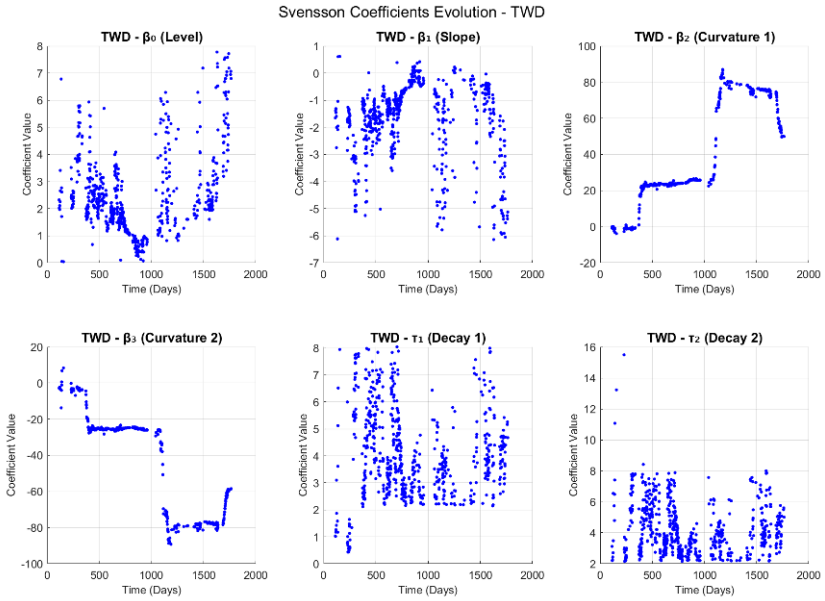


Figure A.69: Svensson model coefficients – TWD

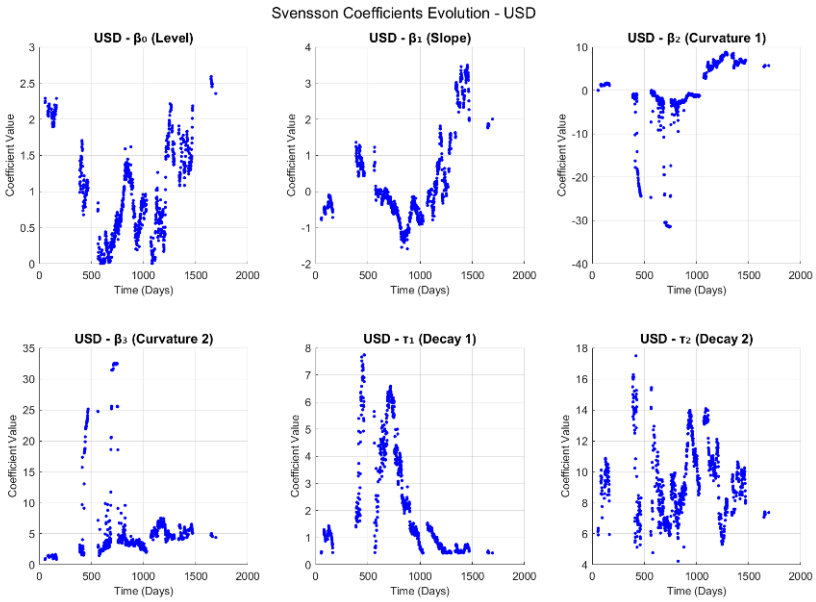


Figure A.70: Svensson model coefficients – USD

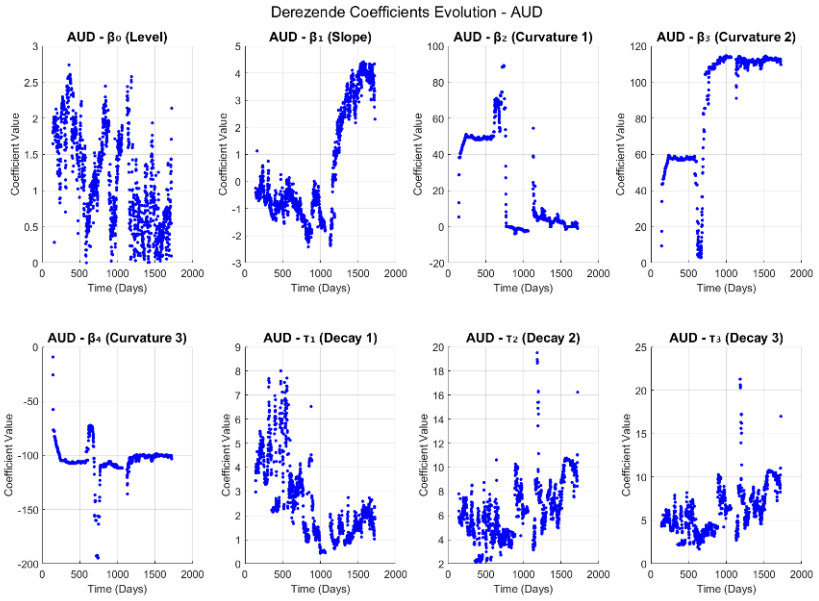


Figure A.71: De Rezende-Ferreira model coefficients – AUD

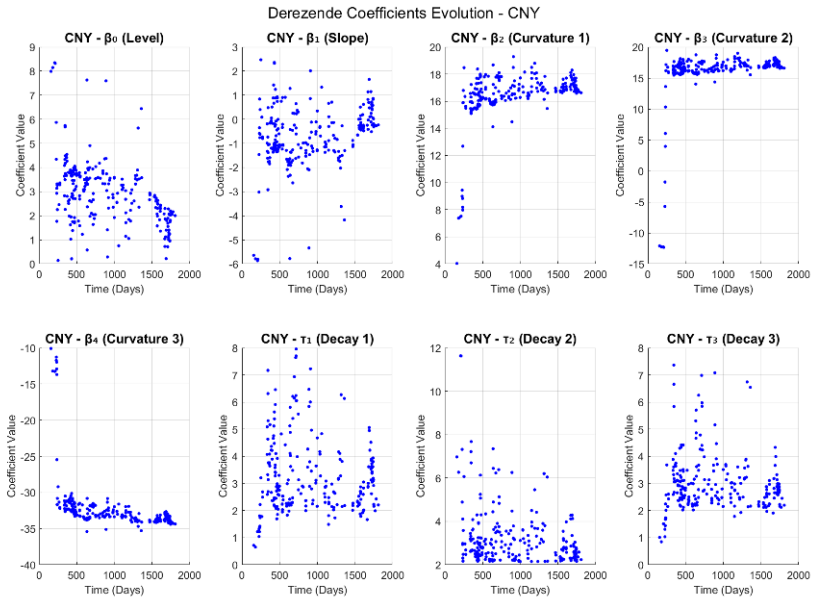


Figure A.72: De Rezende-Ferreira model coefficients – CNY

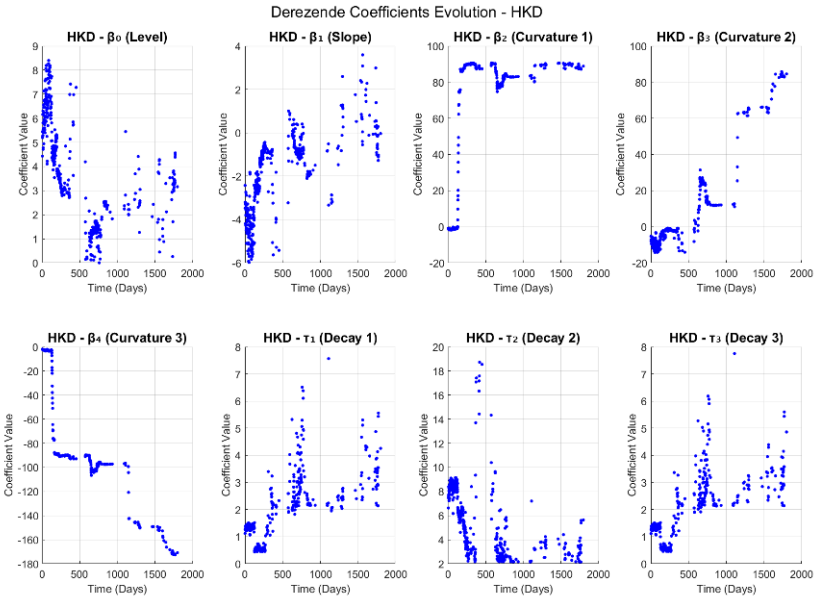


Figure A.73: De Rezende-Ferreira model coefficients – HKD

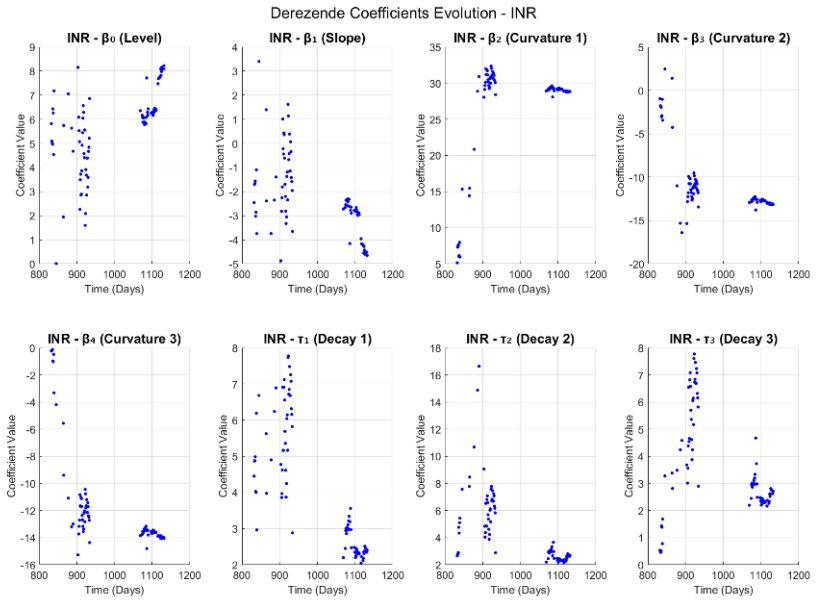


Figure A.74: De Rezende-Ferreira model coefficients – INR

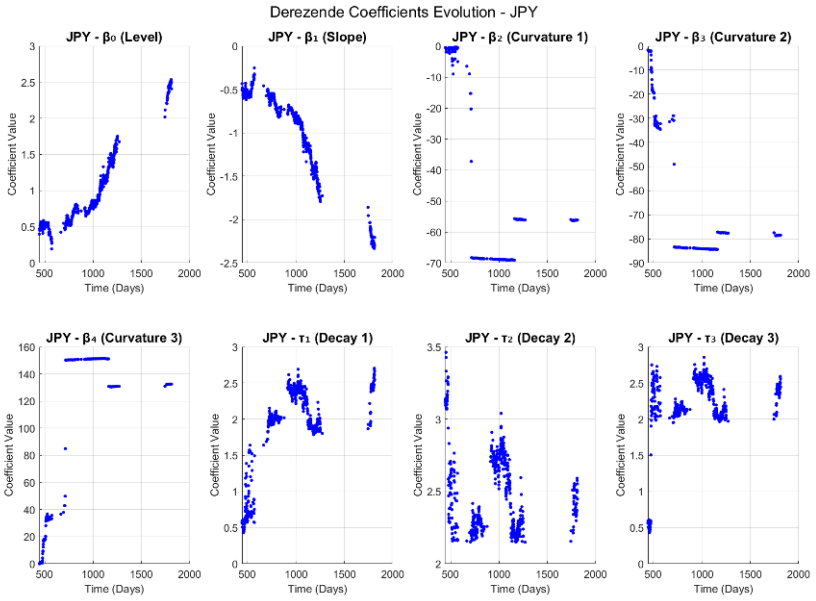


Figure A.75: De Rezende-Ferreira model coefficients – JPY

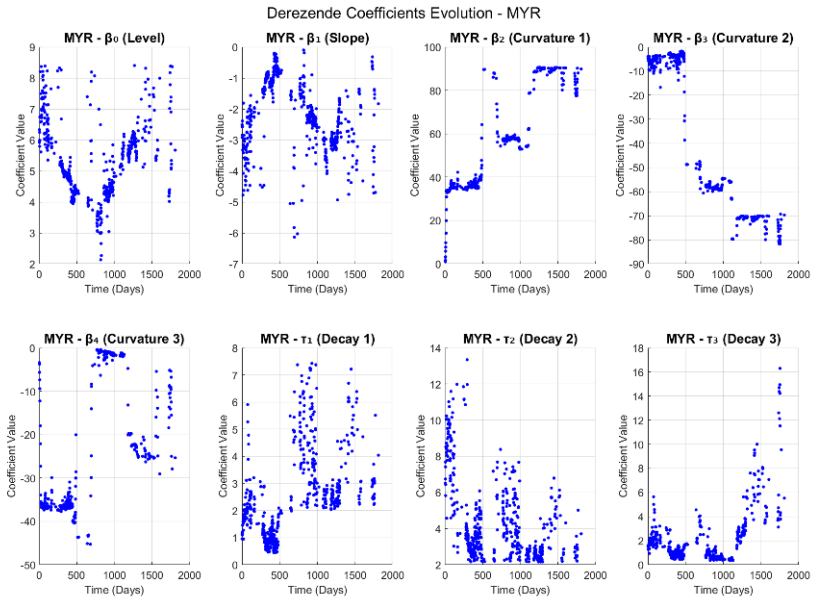


Figure A.76: De Rezende-Ferreira model coefficients – MYR

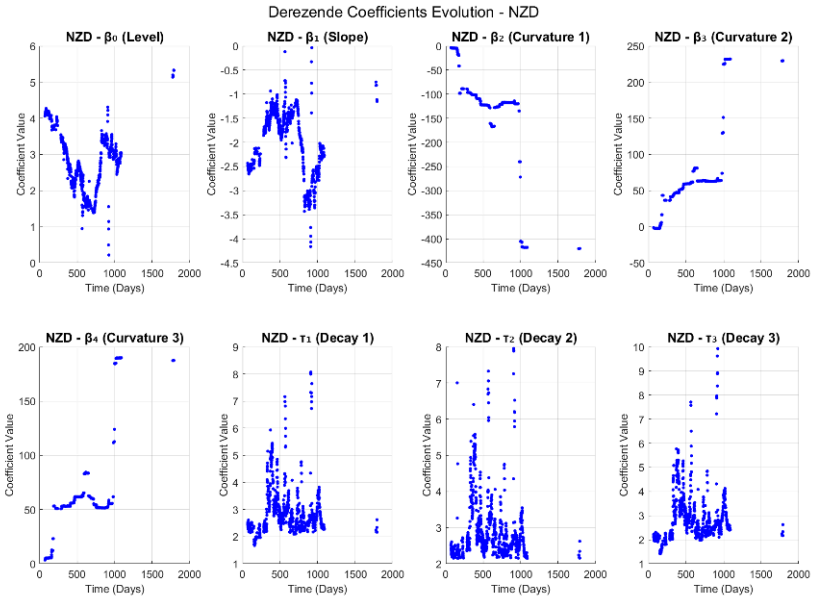


Figure A.77: De Rezende-Ferreira model coefficients – NZD

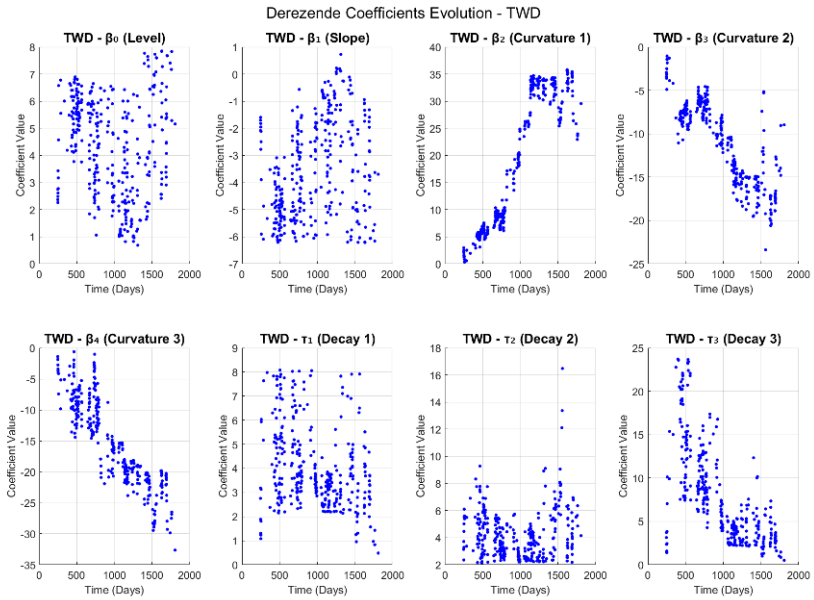


Figure A.78: De Rezende-Ferreira model coefficients – TWD

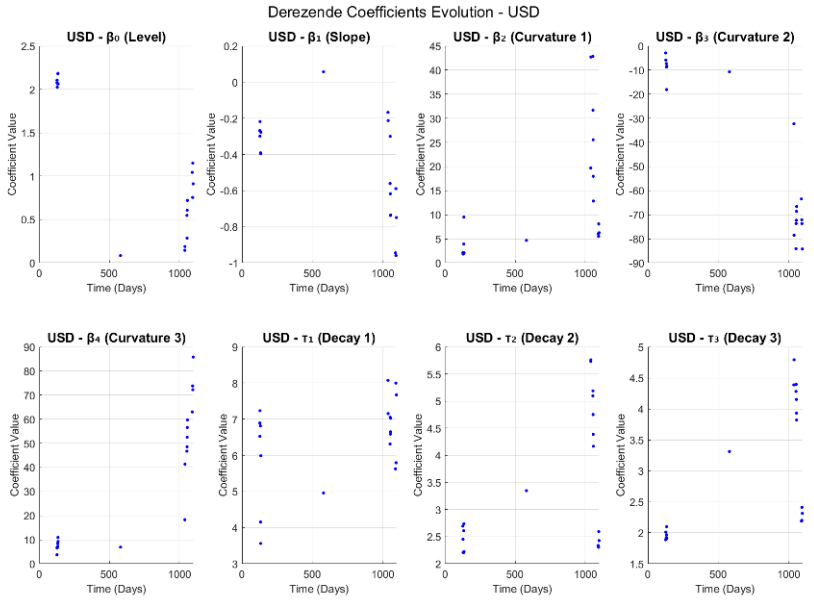


Figure A.79: De Rezende-Ferreira model coefficients – USD

A.2.3 Optimized Models (Pacific Economies)

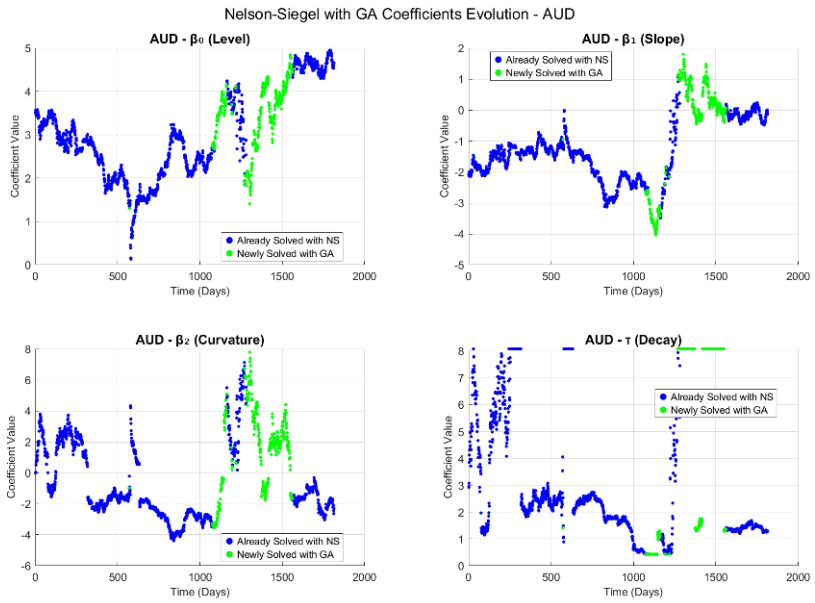


Figure A.80: Nelson-Siegel model (GA-optimized) coefficients – AUD

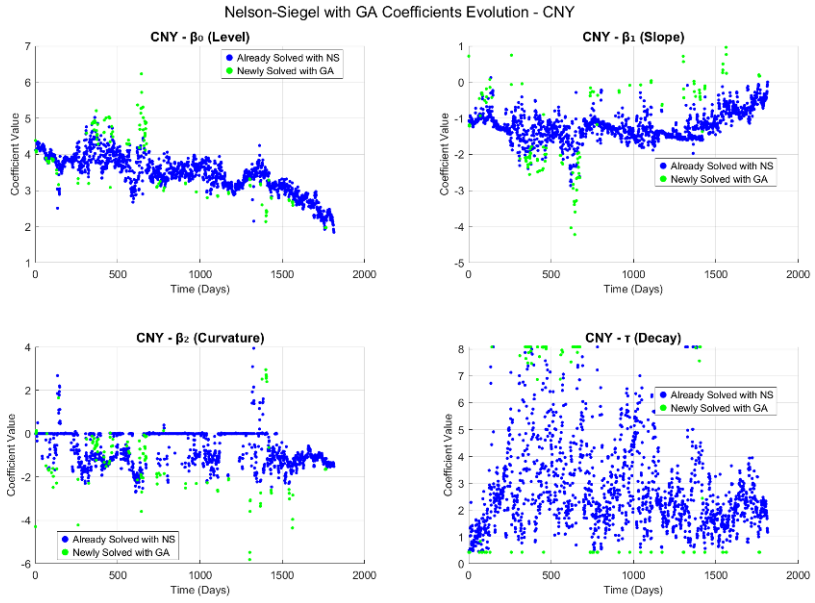


Figure A.81: Nelson-Siegel model (GA-optimized) coefficients – CNY

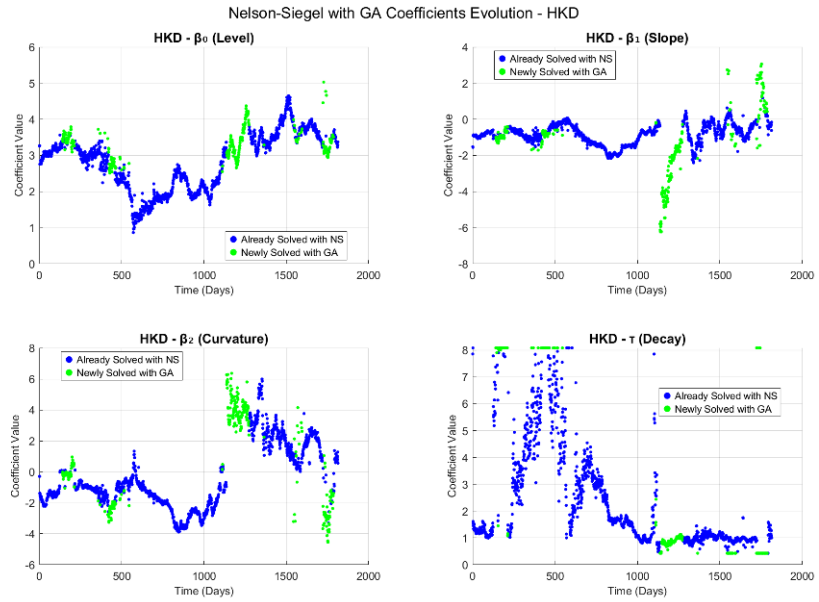


Figure A.82: Nelson-Siegel model (GA-optimized) coefficients – HKD

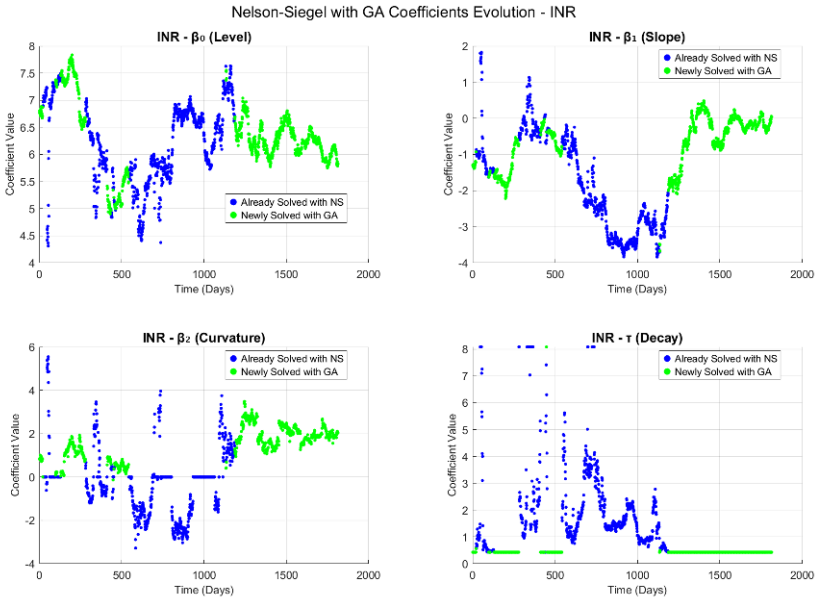


Figure A.83: Nelson-Siegel model (GA-optimized) coefficients – INR

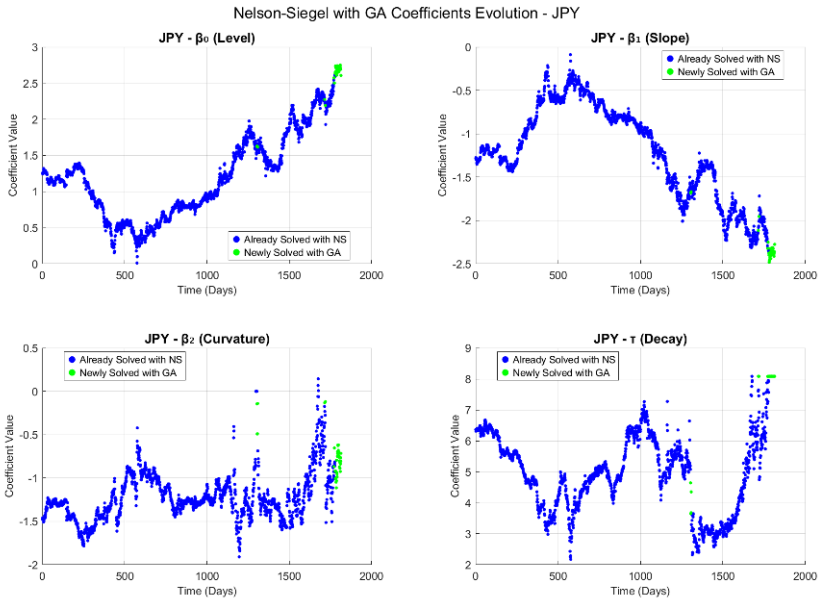


Figure A.84: Nelson-Siegel model (GA-optimized) coefficients – JPY

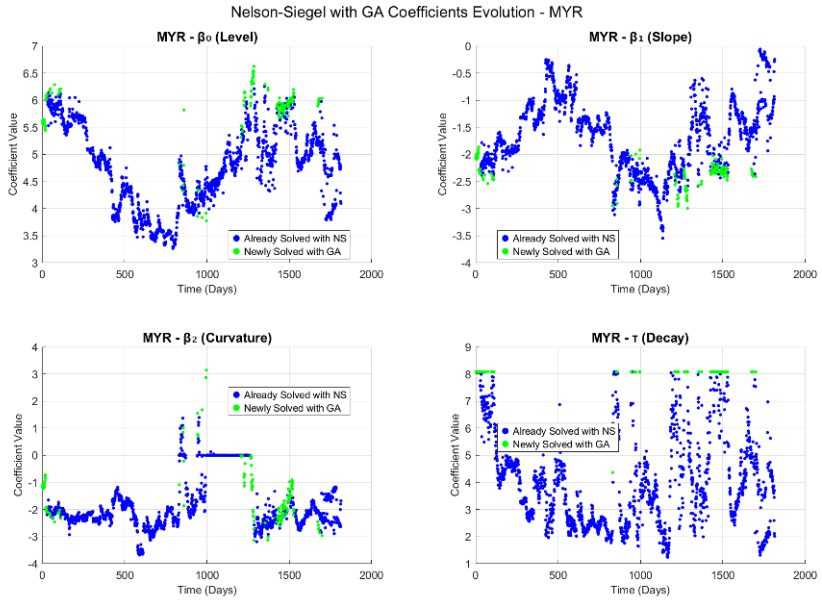


Figure A.85: Nelson-Siegel model (GA-optimized) coefficients – MYR

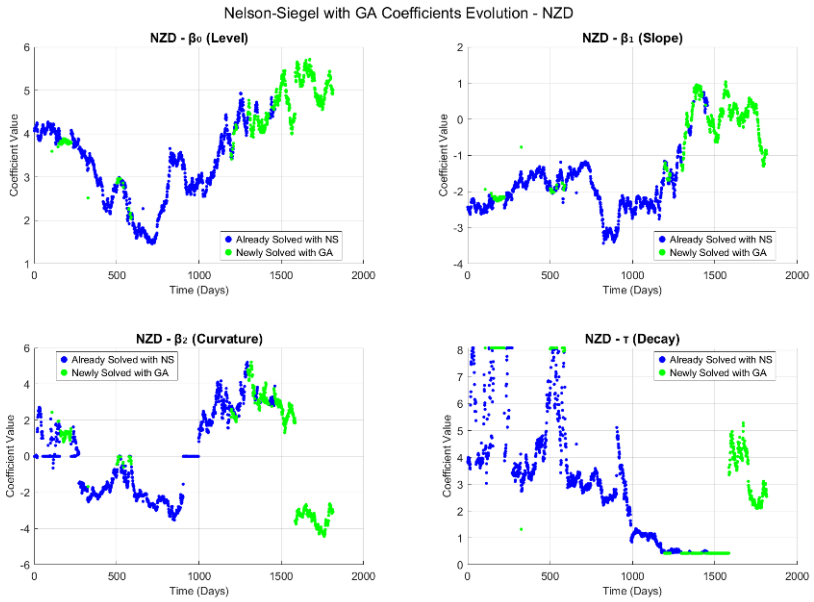


Figure A.86: Nelson-Siegel model (GA-optimized) coefficients – NZD

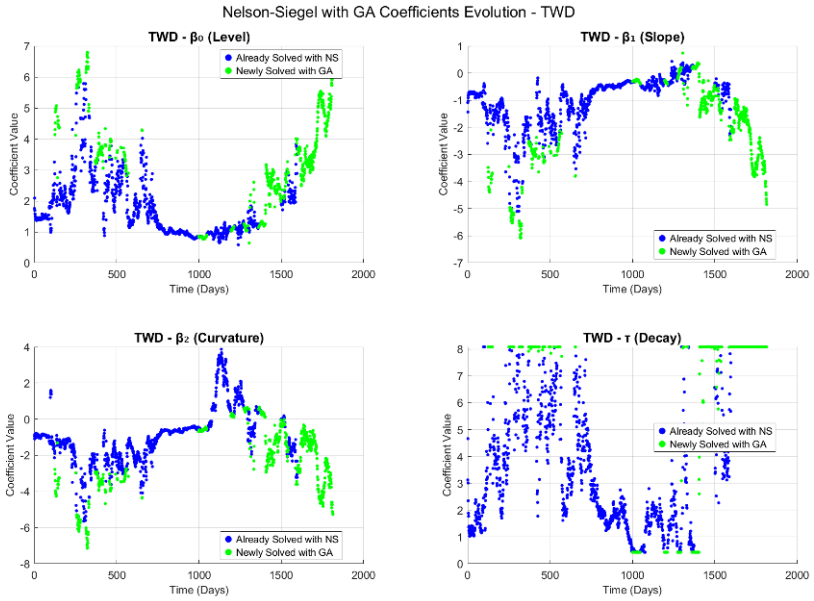


Figure A.87: Nelson-Siegel model (GA-optimized) coefficients – TWD

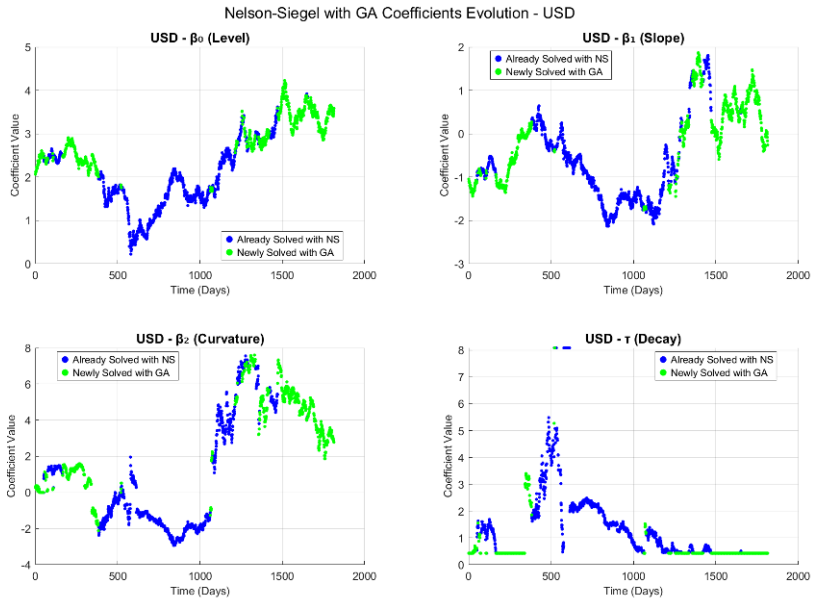


Figure A.88: Nelson-Siegel model (GA-optimized) coefficients – USD

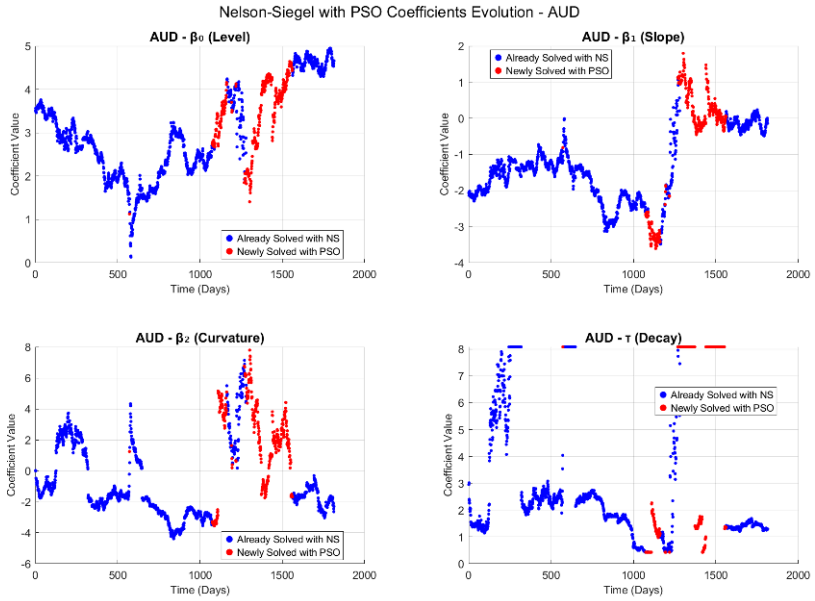


Figure A.89: Nelson-Siegel model (PSO-optimized) coefficients – AUD

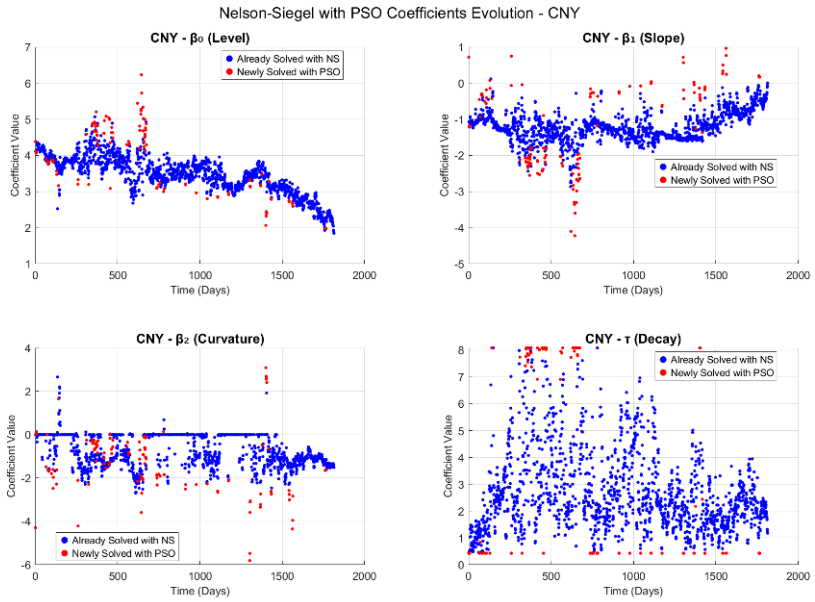


Figure A.90: Nelson-Siegel model (PSO-optimized) coefficients – CNY

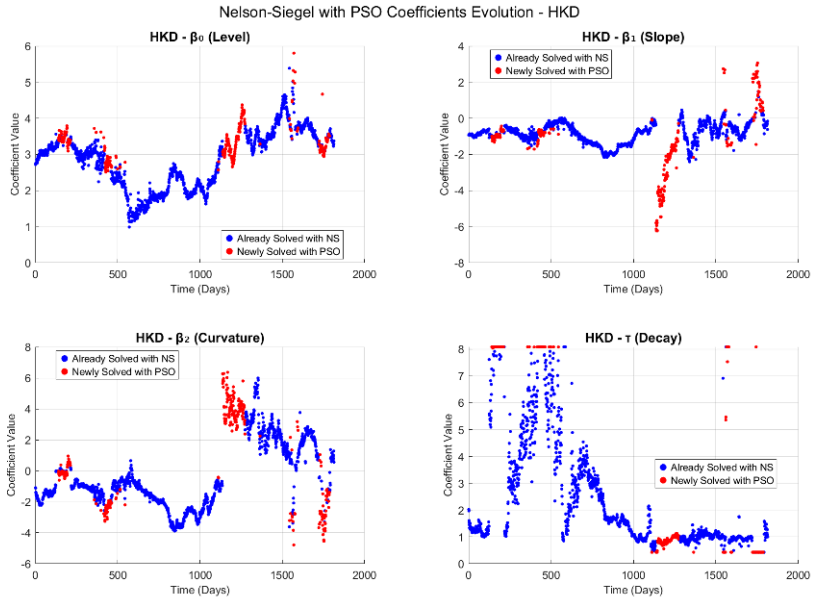


Figure A.91: Nelson-Siegel model (PSO-optimized) coefficients – HKD

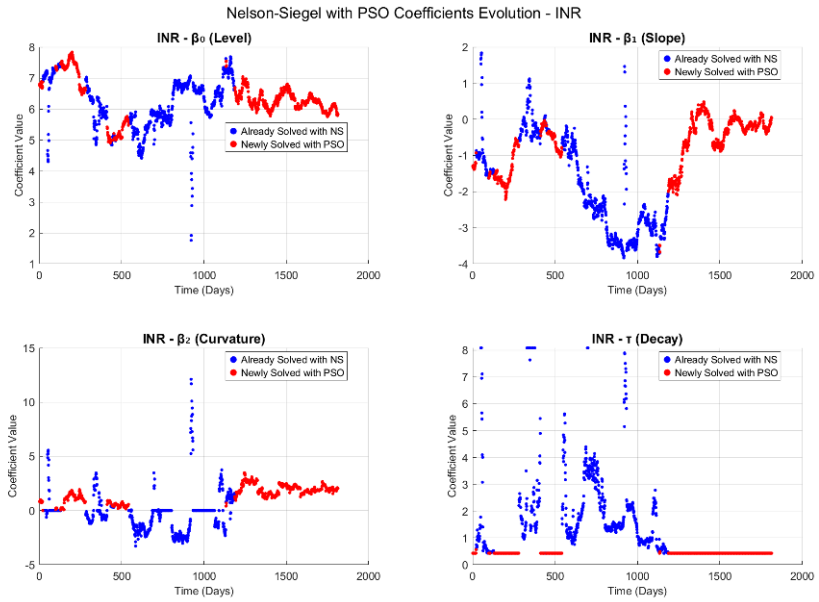


Figure A.92: Nelson-Siegel model (PSO-optimized) coefficients – INR

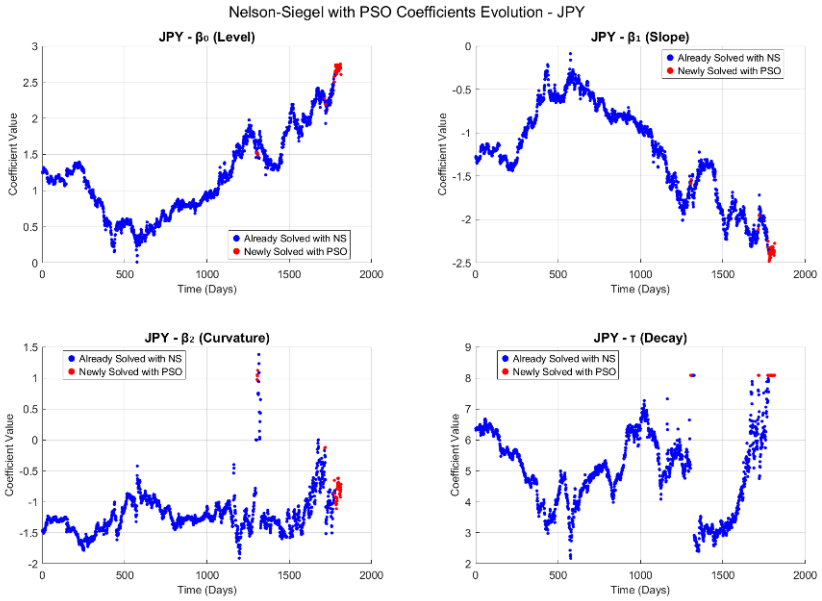


Figure A.93: Nelson-Siegel model (PSO-optimized) coefficients – JPY

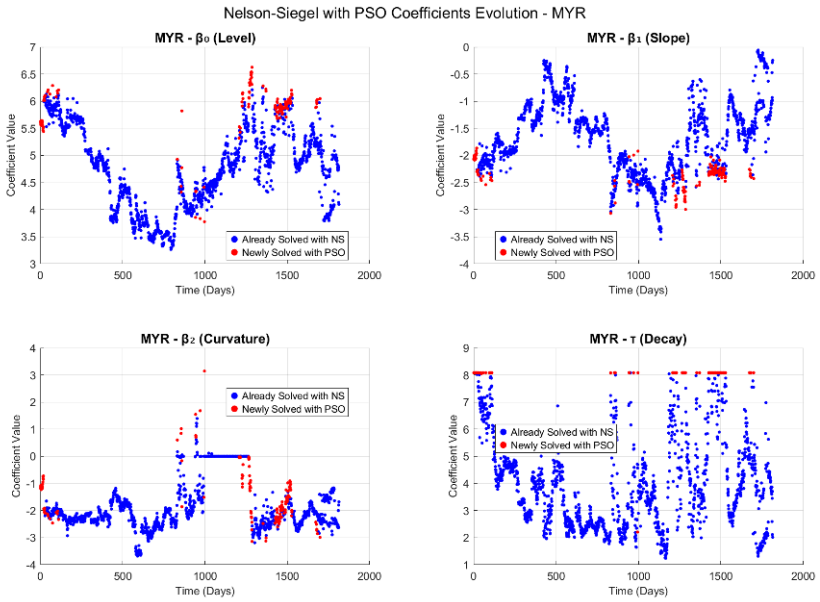


Figure A.94: Nelson-Siegel model (PSO-optimized) coefficients – MYR

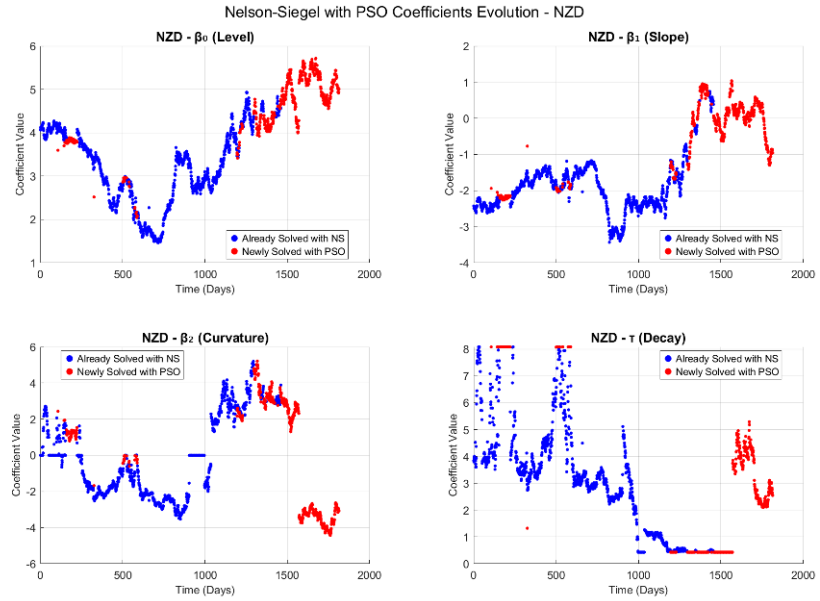


Figure A.95: Nelson-Siegel model (PSO-optimized) coefficients – NZD

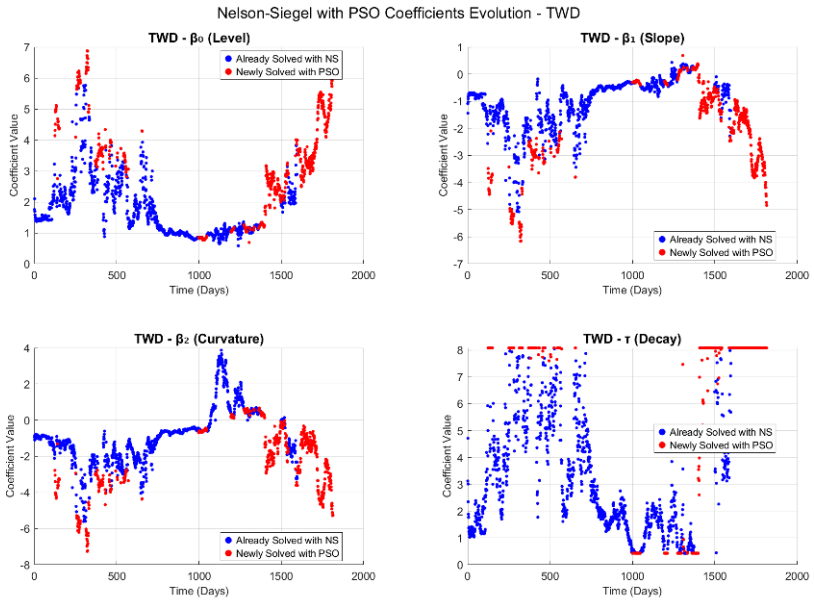


Figure A.96: Nelson-Siegel model (PSO-optimized) coefficients – TWD

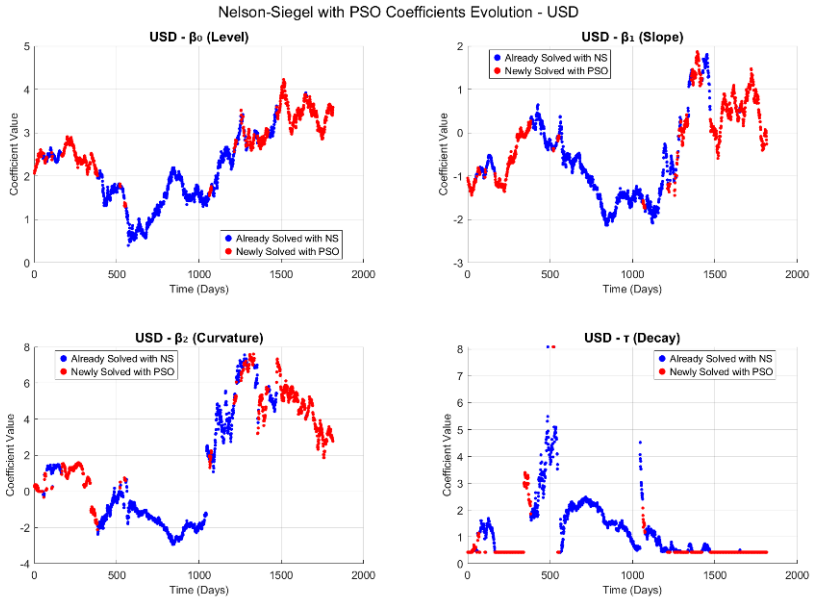


Figure A.97: Nelson-Siegel model (PSO-optimized) coefficients – USD

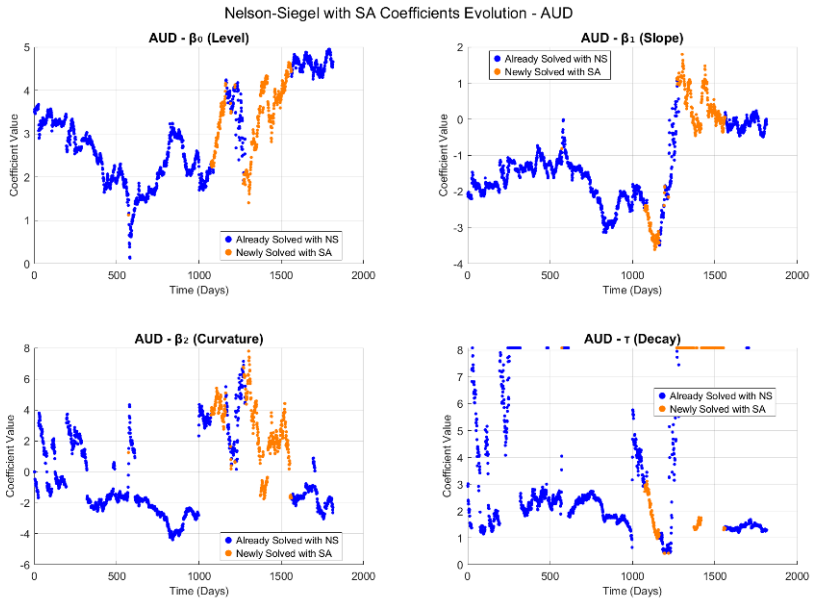


Figure A.98: Nelson-Siegel model (SA-optimized) coefficients – AUD

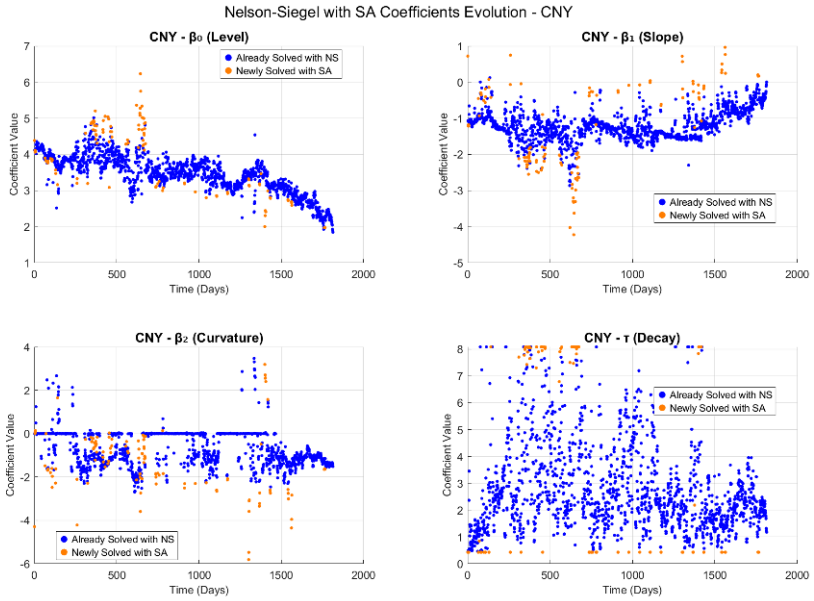


Figure A.99: Nelson-Siegel model (SA-optimized) coefficients – CNY

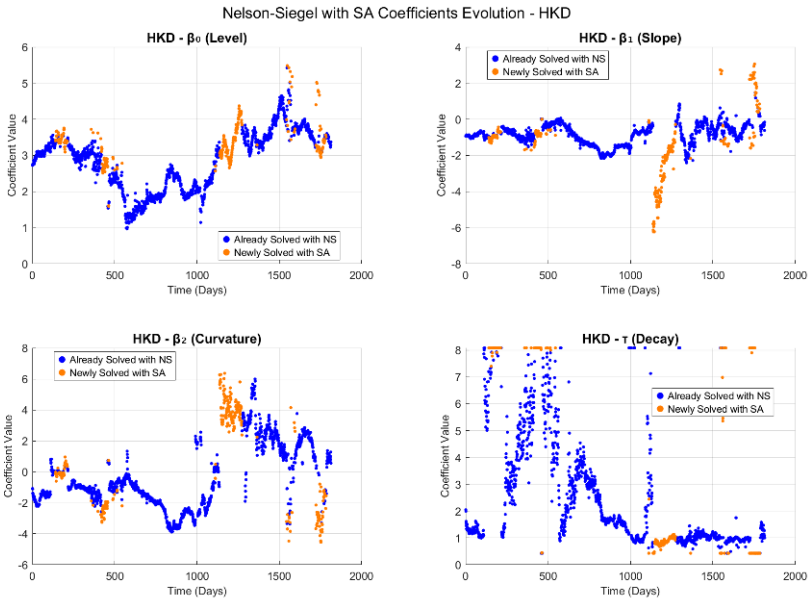


Figure A.100: Nelson-Siegel model (SA-optimized) coefficients – HKD

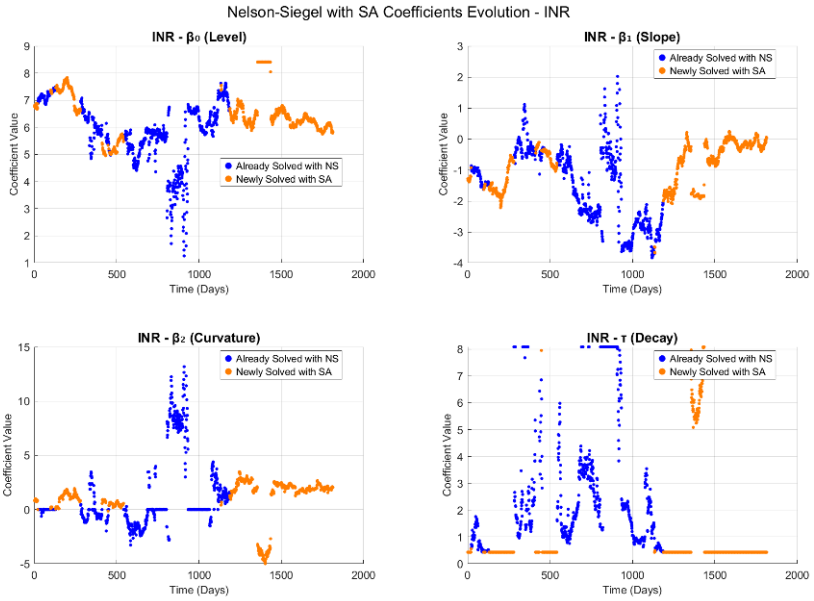


Figure A.101: Nelson-Siegel model (SA-optimized) coefficients – INR

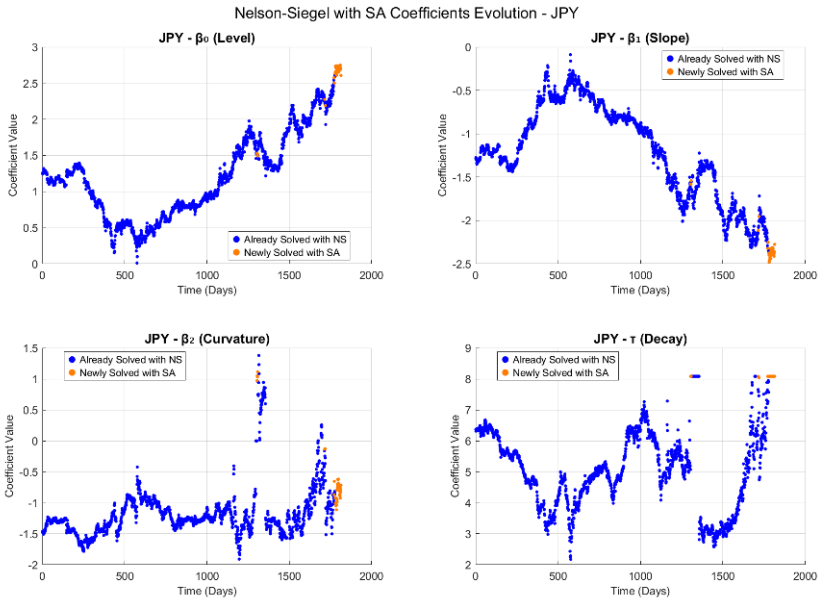


Figure A.102: Nelson-Siegel model (SA-optimized) coefficients – JPY

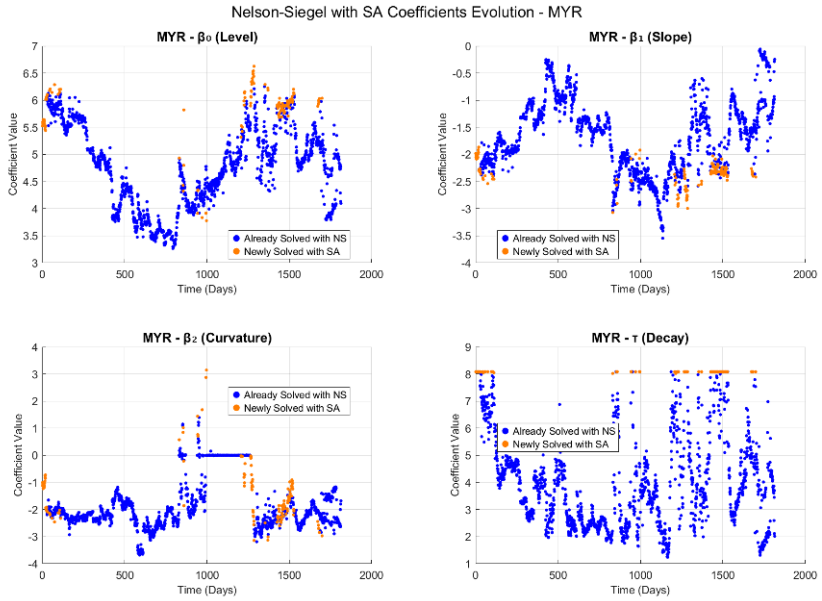


Figure A.103: Nelson-Siegel model (SA-optimized) coefficients – MYR

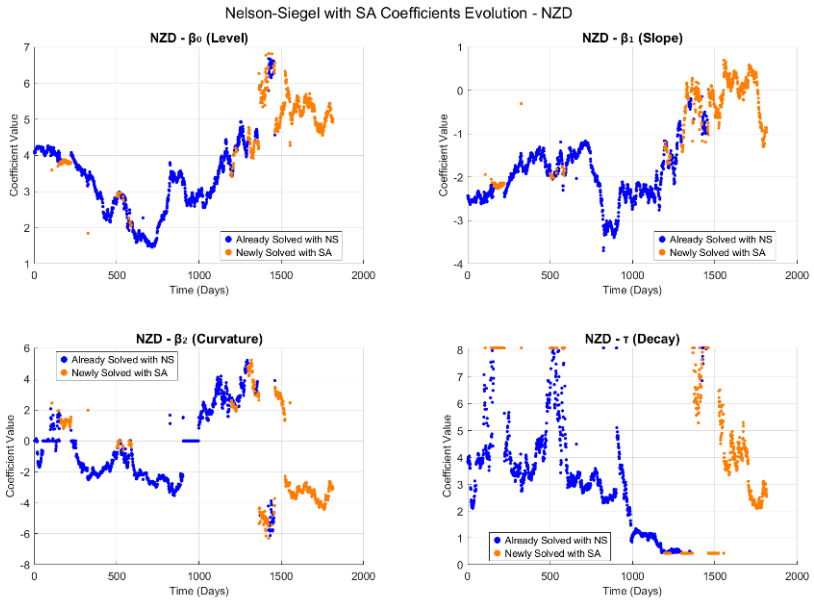


Figure A.104: Nelson-Siegel model (SA-optimized) coefficients – NZD

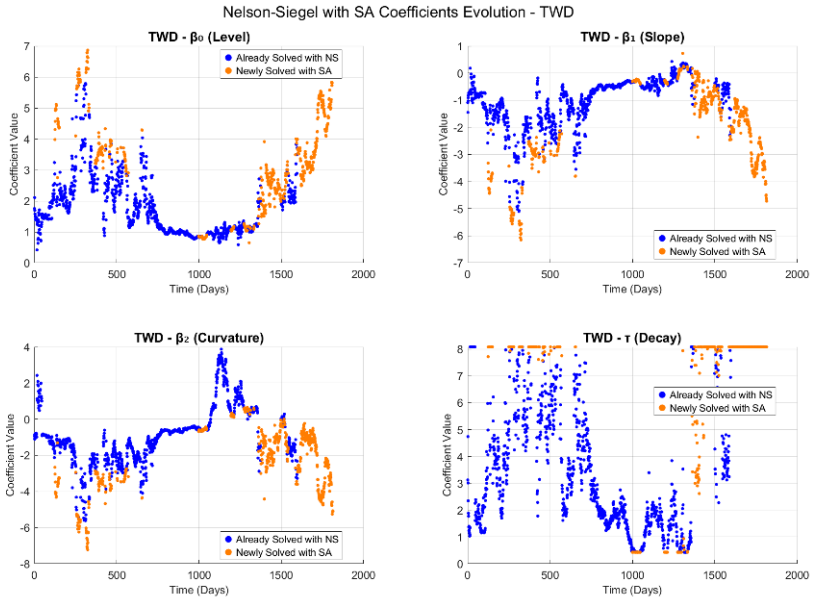


Figure A.105: Nelson-Siegel model (SA-optimized) coefficients – TWD

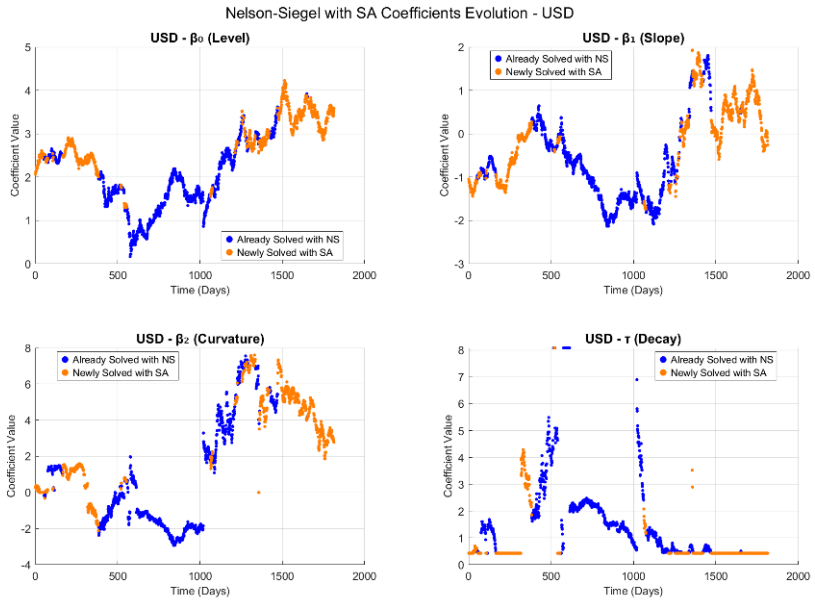


Figure A.106: Nelson-Siegel model (SA-optimized) coefficients – USD

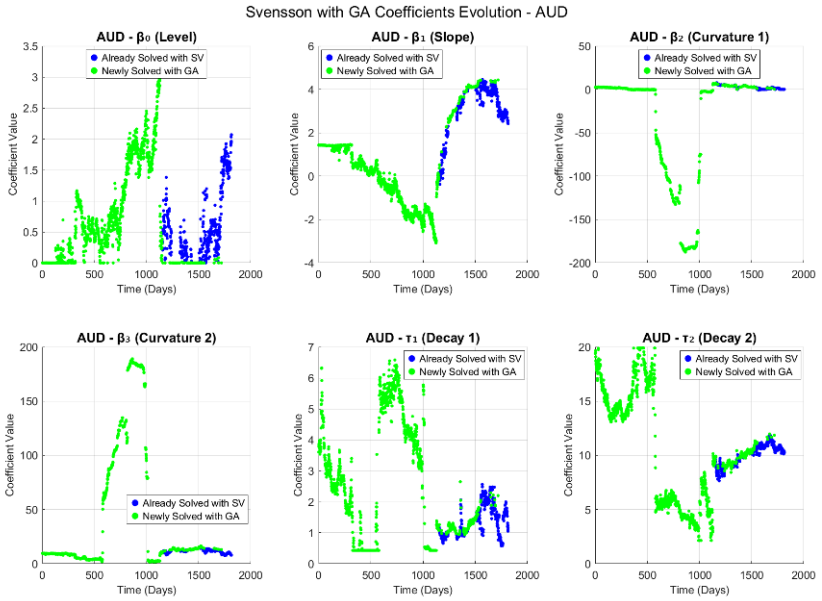


Figure A.107: Svensson model (GA-optimized) coefficients – AUD

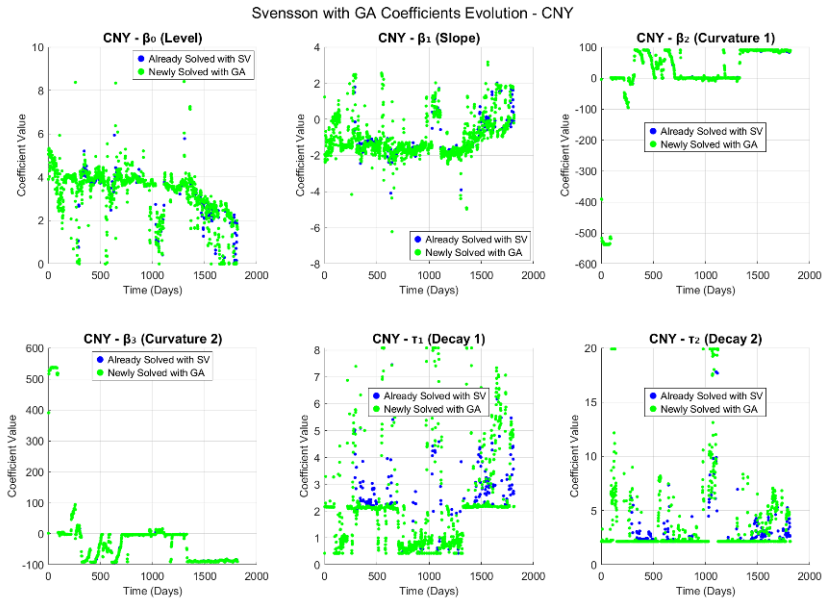


Figure A.108: Svensson model (GA-optimized) coefficients – CNY

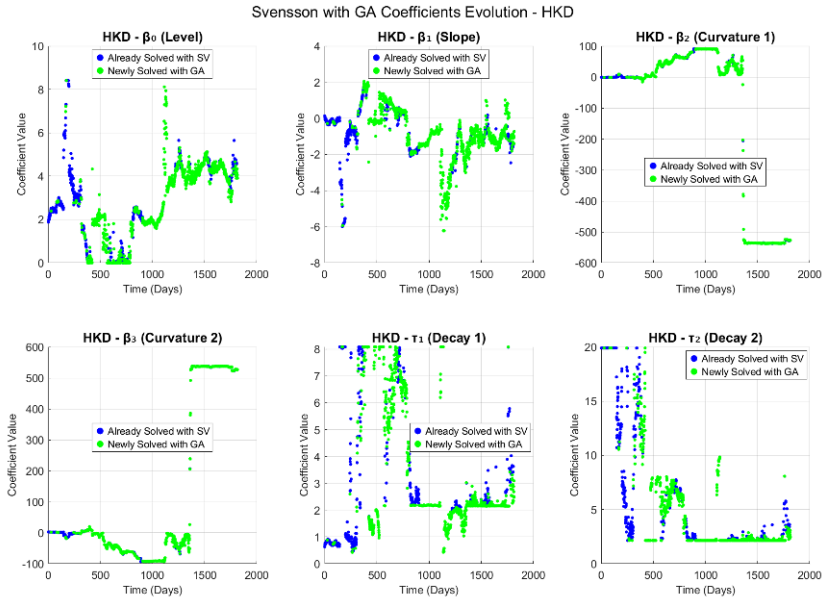


Figure A.109: Svensson model (GA-optimized) coefficients – HKD

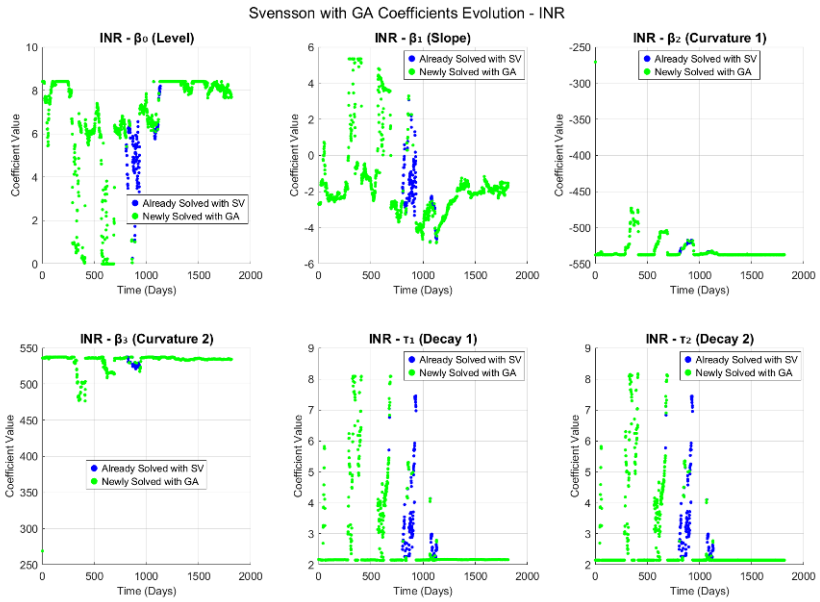


Figure A.110: Svensson model (GA-optimized) coefficients – INR

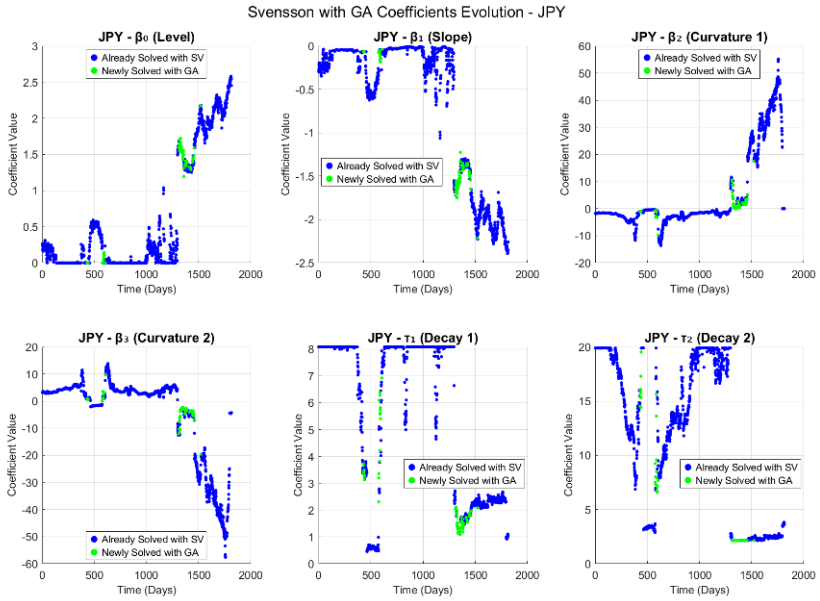


Figure A.111: Svensson model (GA-optimized) coefficients – JPY

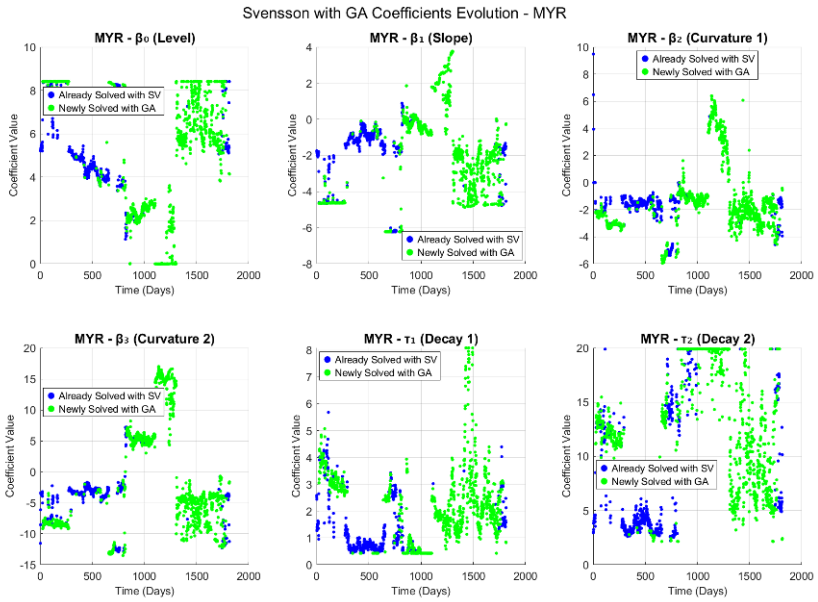


Figure A.112: Svensson model (GA-optimized) coefficients – MYR

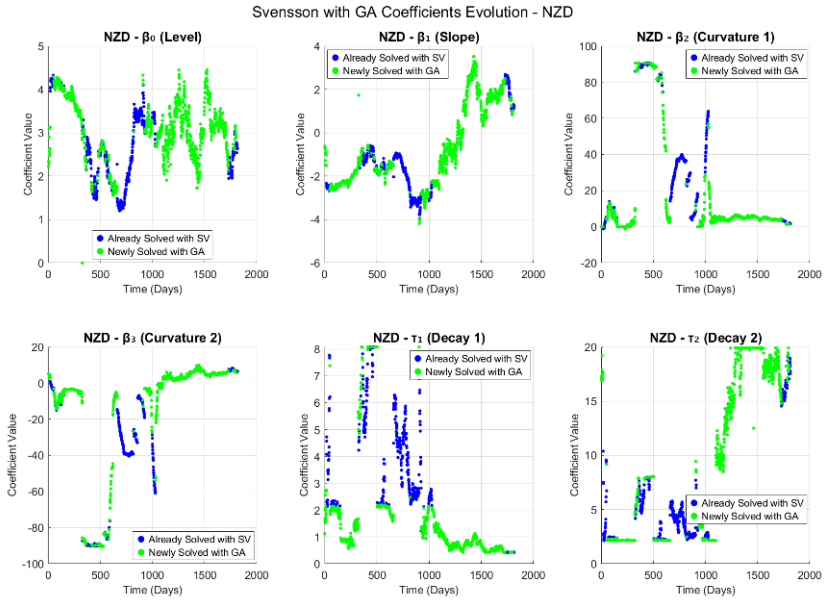


Figure A.113: Svensson model (GA-optimized) coefficients – NZD

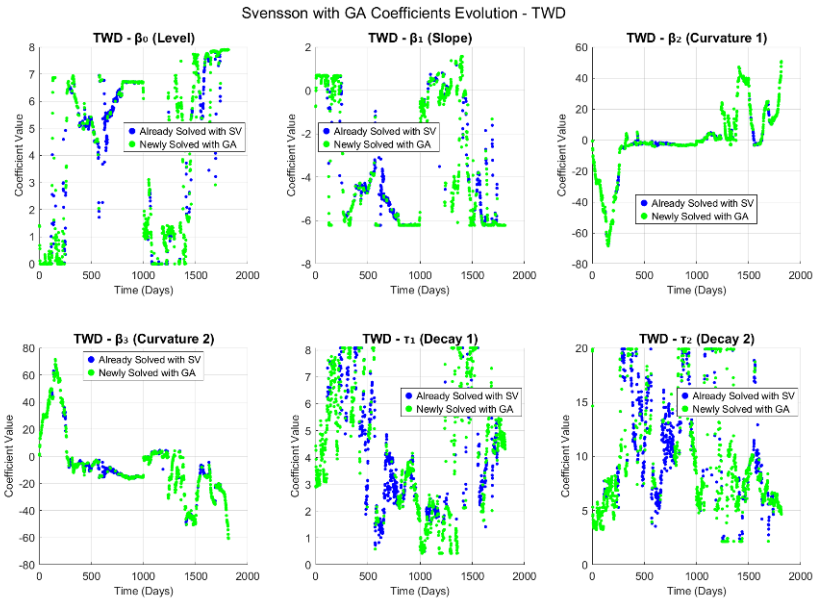


Figure A.114: Svensson model (GA-optimized) coefficients – TWD

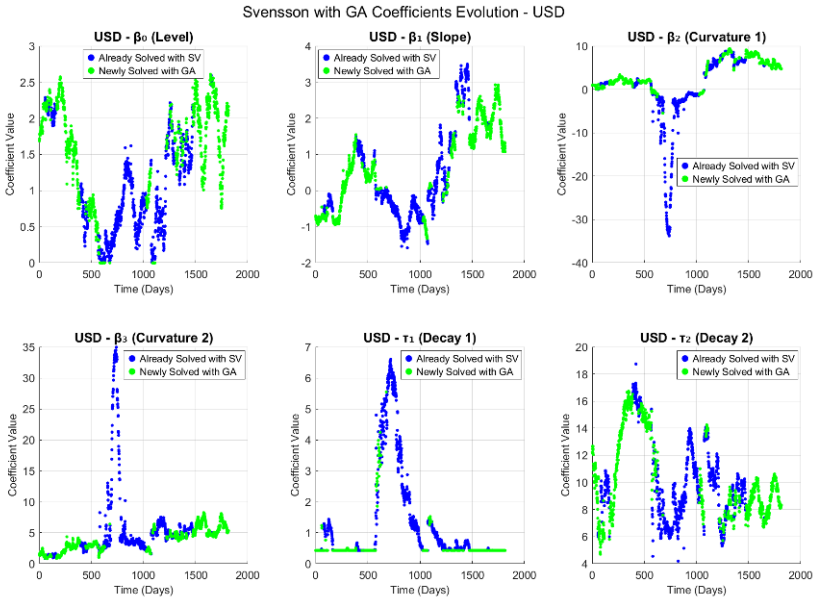


Figure A.115: Svensson model (GA-optimized) coefficients – USD

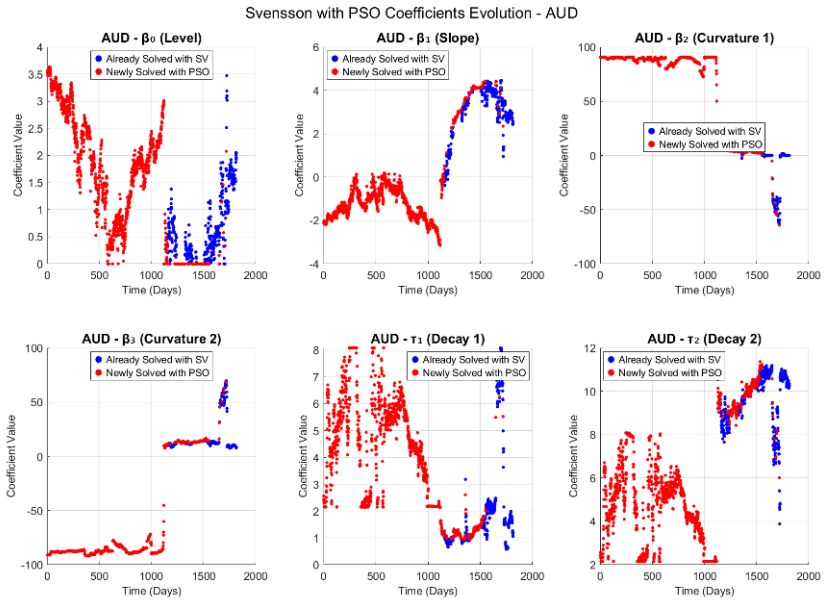


Figure A.116: Svensson model (PSO-optimized) coefficients – AUD

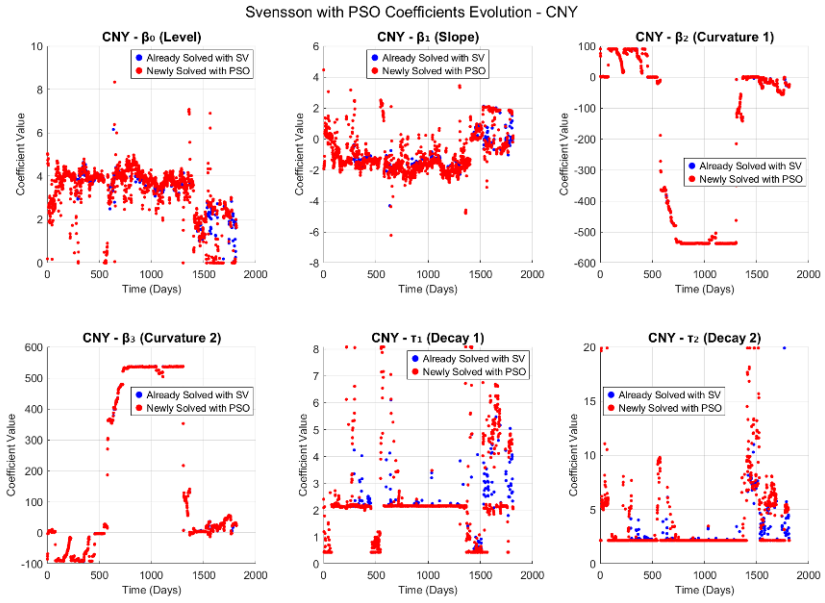


Figure A.117: Svensson model (PSO-optimized) coefficients – CNY

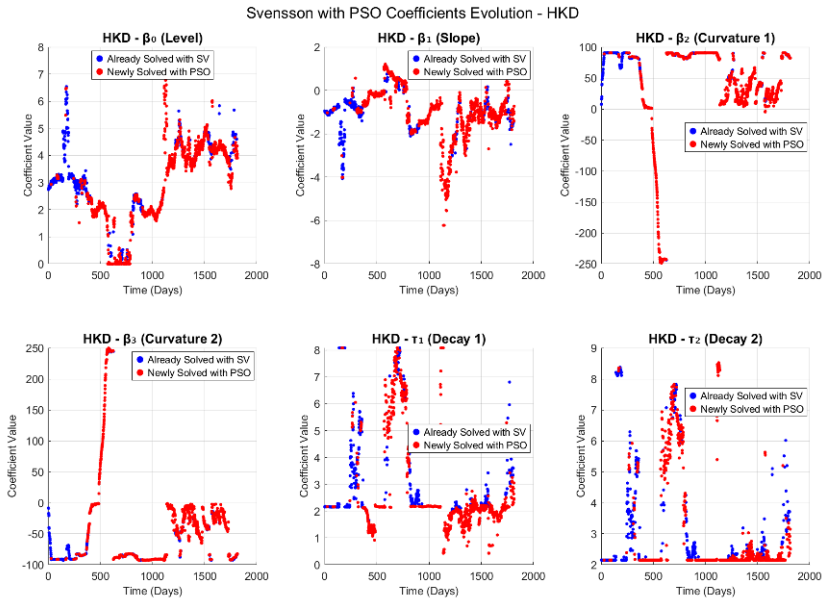


Figure A.118: Svensson model (PSO-optimized) coefficients – HKD

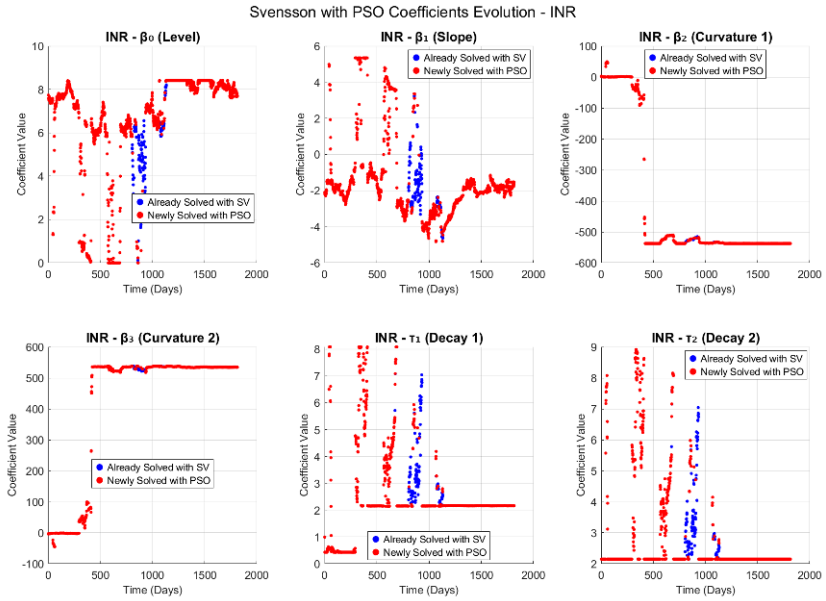


Figure A.119: Svensson model (PSO-optimized) coefficients – INR

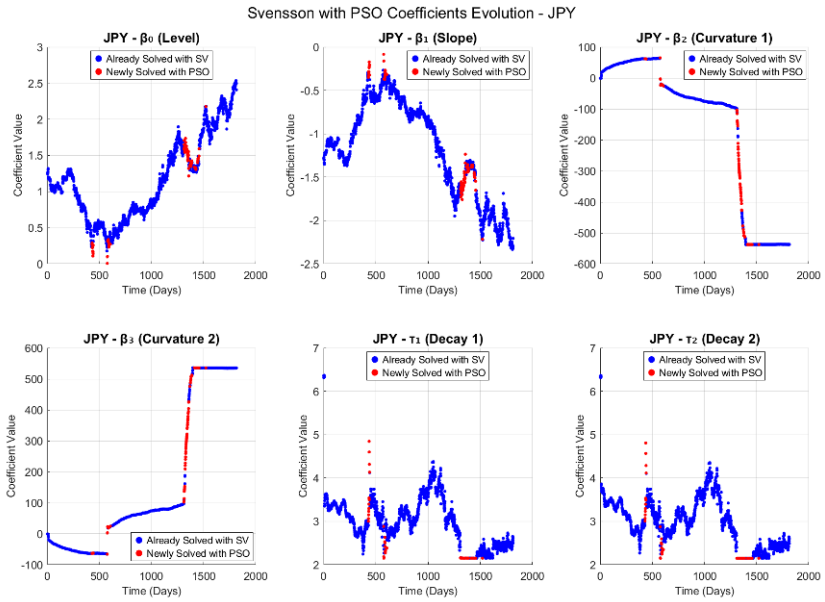


Figure A.120: Svensson model (PSO-optimized) coefficients – JPY

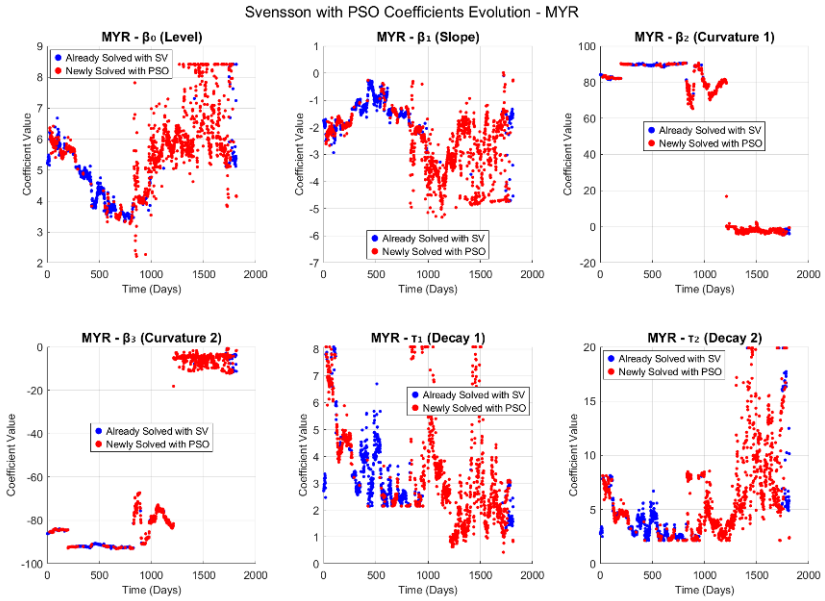


Figure A.121: Svensson model (PSO-optimized) coefficients – MYR

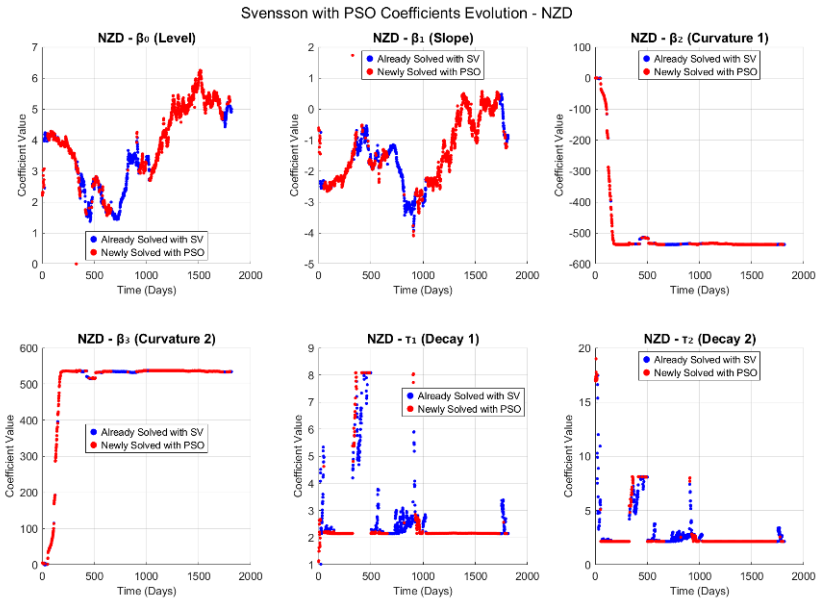


Figure A.122: Svensson model (PSO-optimized) coefficients – NZD

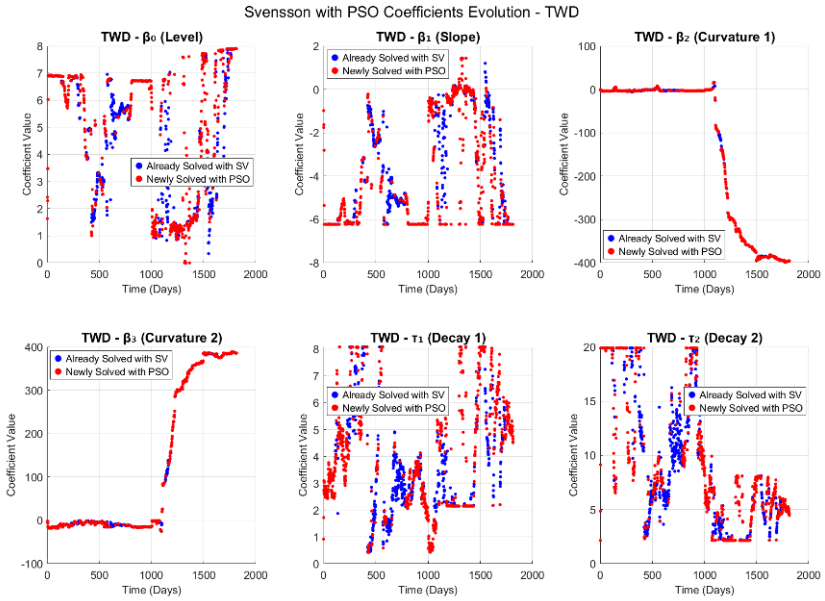


Figure A.123: Svensson model (PSO-optimized) coefficients – TWD

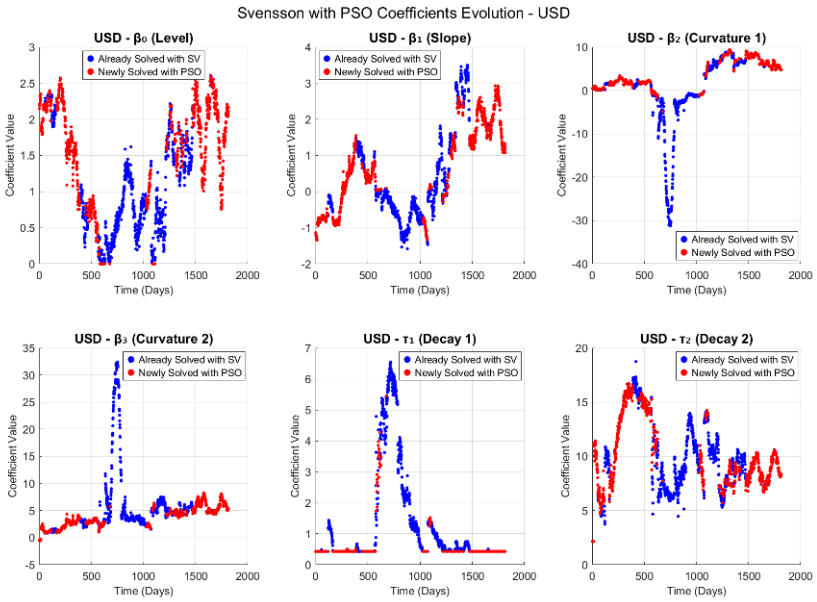


Figure A.124: Svensson model (PSO-optimized) coefficients – USD

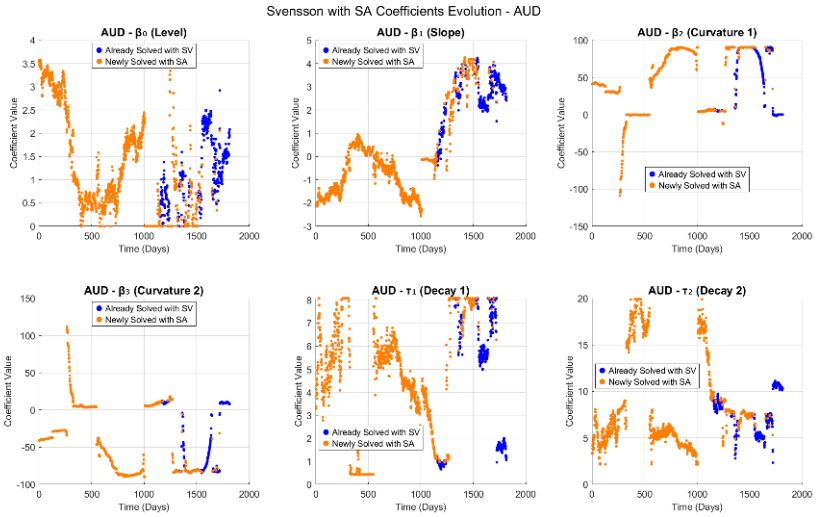


Figure A.125: Svensson model (SA-optimized) coefficients – AUD

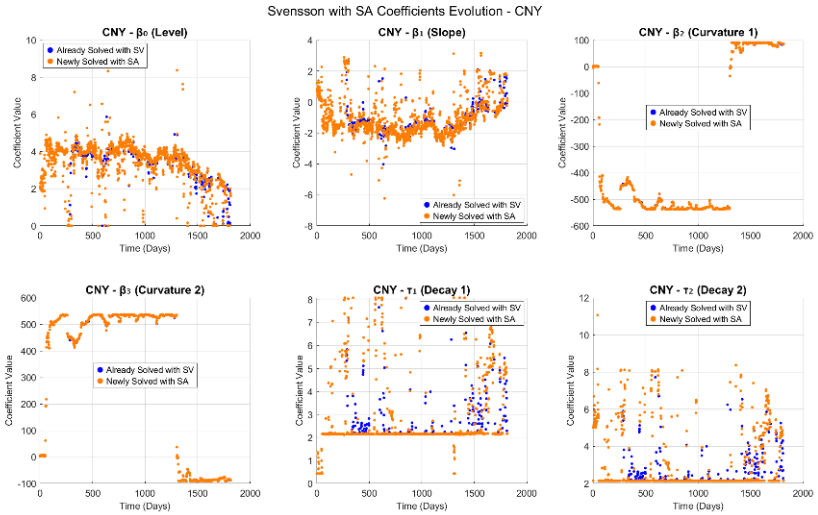


Figure A.126: Svensson model (SA-optimized) coefficients – CNY

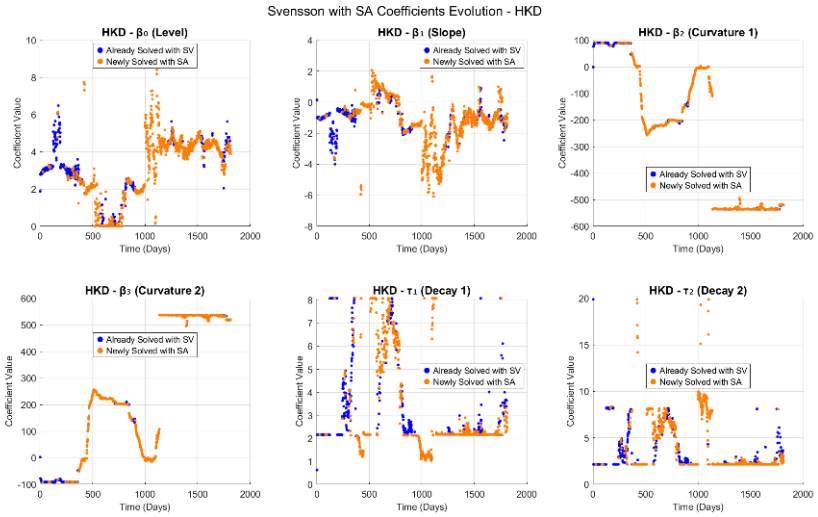


Figure A.127: Svensson model (SA-optimized) coefficients – HKD

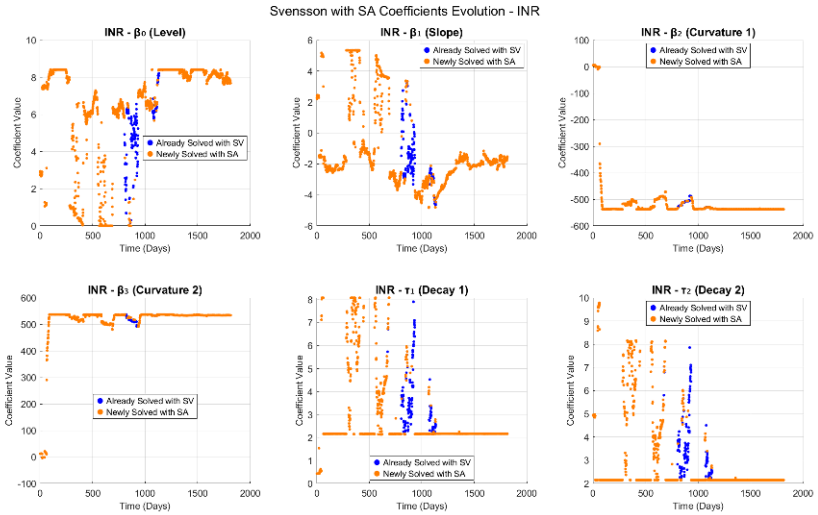


Figure A.128: Svensson model (SA-optimized) coefficients – INR

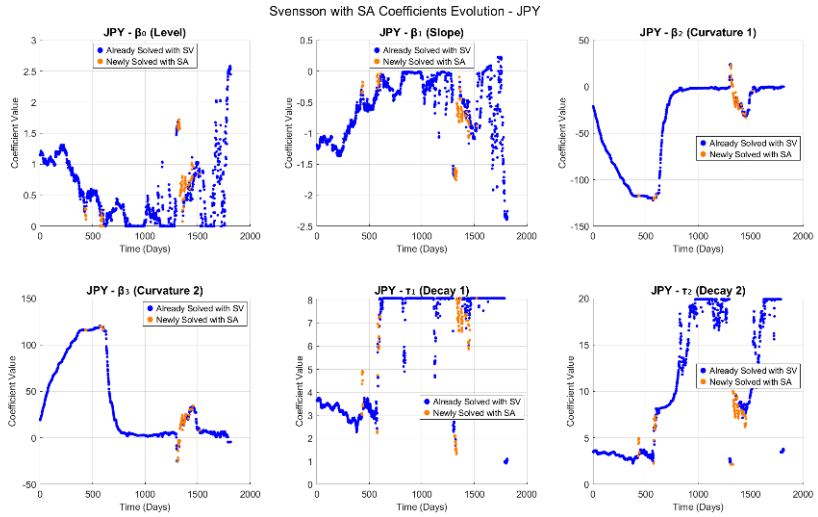


Figure A.129: Svensson model (SA-optimized) coefficients – JPY

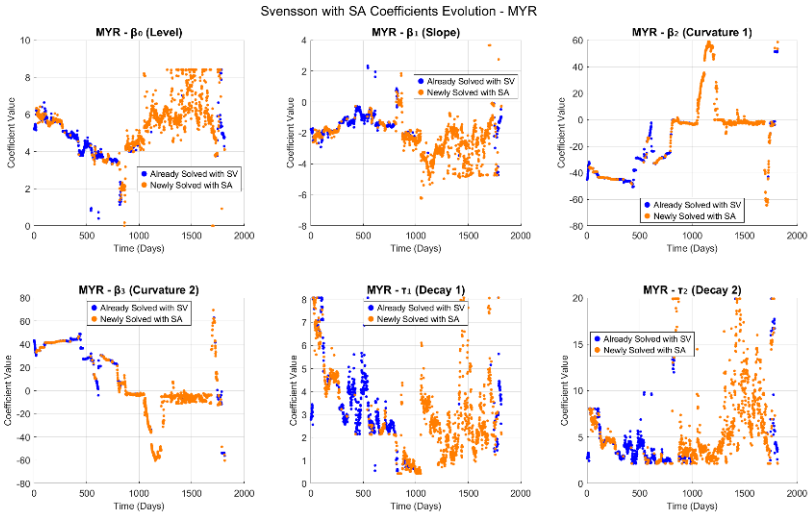


Figure A.130: Svensson model (SA-optimized) coefficients – MYR

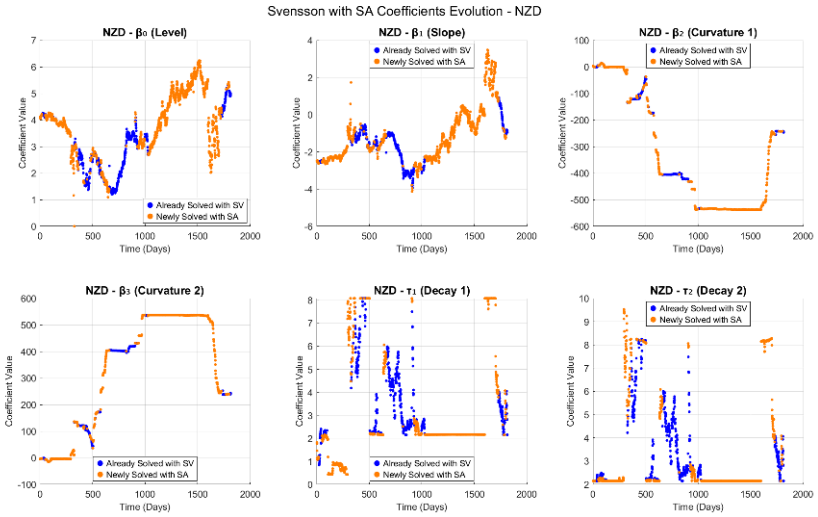


Figure A.131: Svensson model (SA-optimized) coefficients – NZD

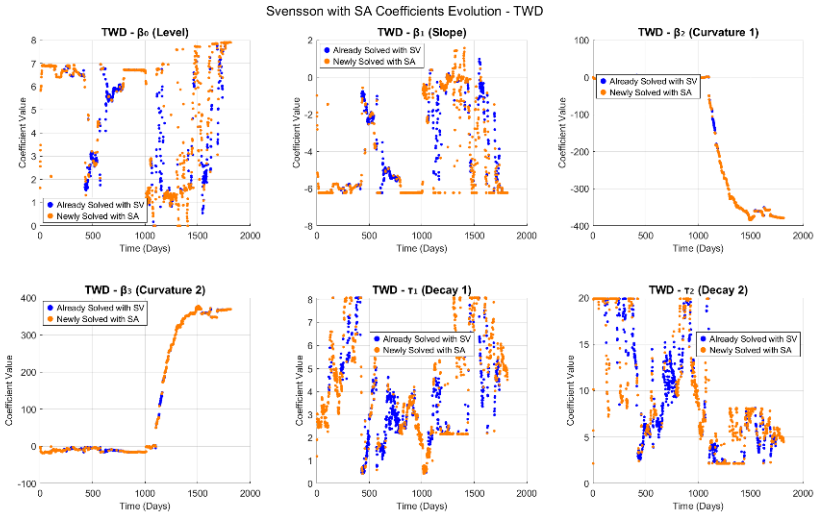


Figure A.132: Svensson model (SA-optimized) coefficients – TWD

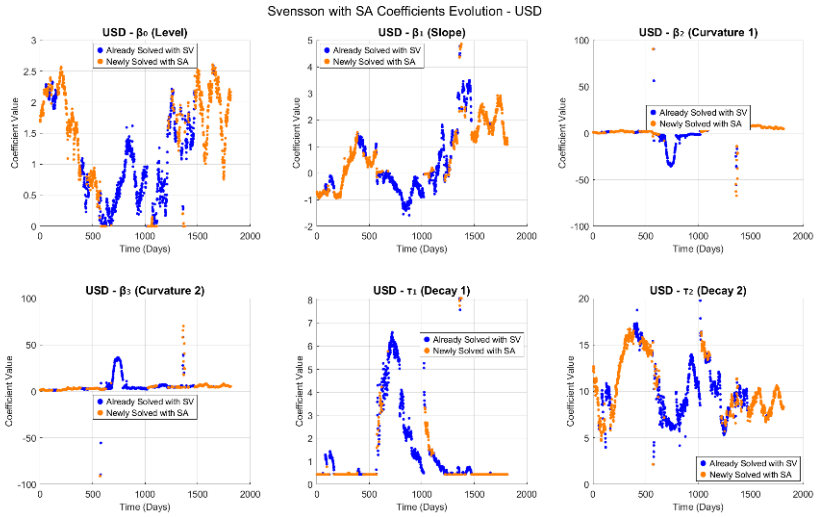


Figure A.133: Svensson model (SA-optimized) coefficients – USD

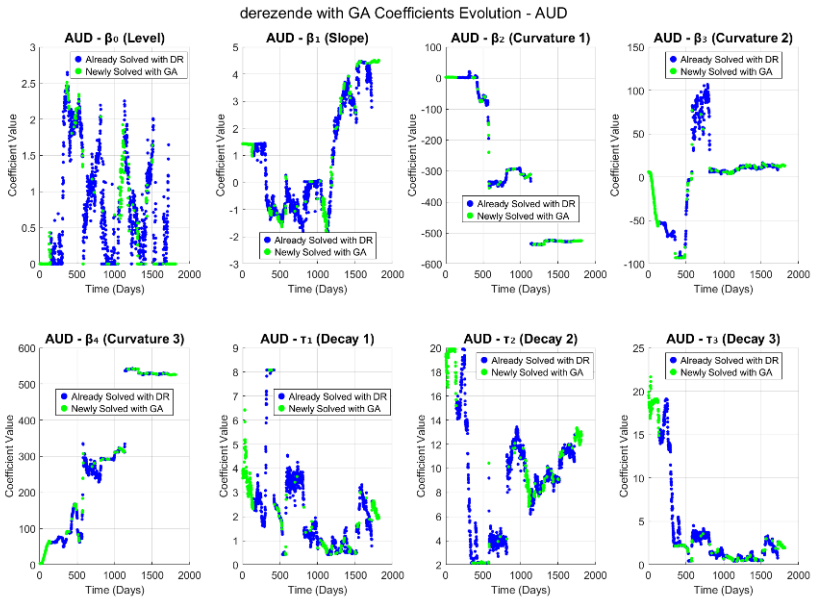


Figure A.134: De Rezende-Ferreira model (GA-optimized) coefficients – AUD

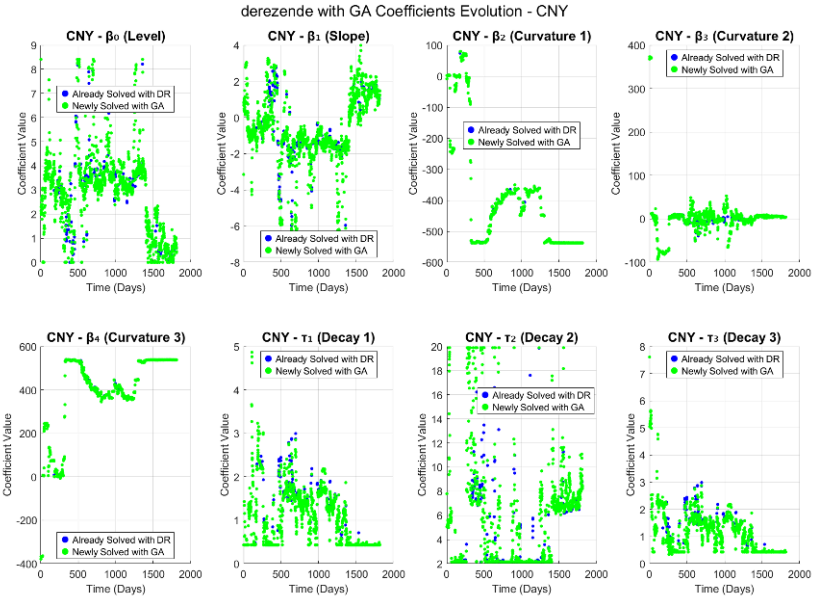


Figure A.135: De Rezende-Ferreira model (GA-optimized) coefficients – CNY

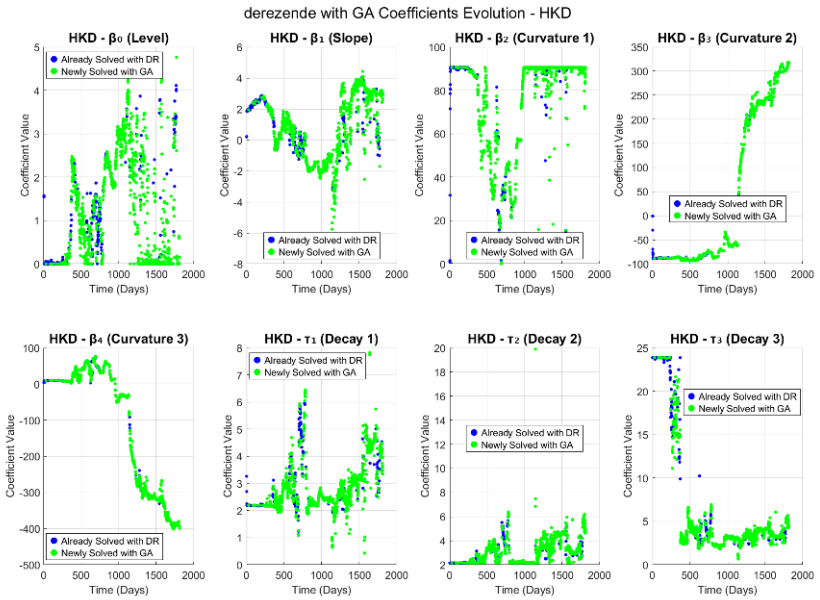


Figure A.136: De Rezende-Ferreira model (GA-optimized) coefficients – HKD

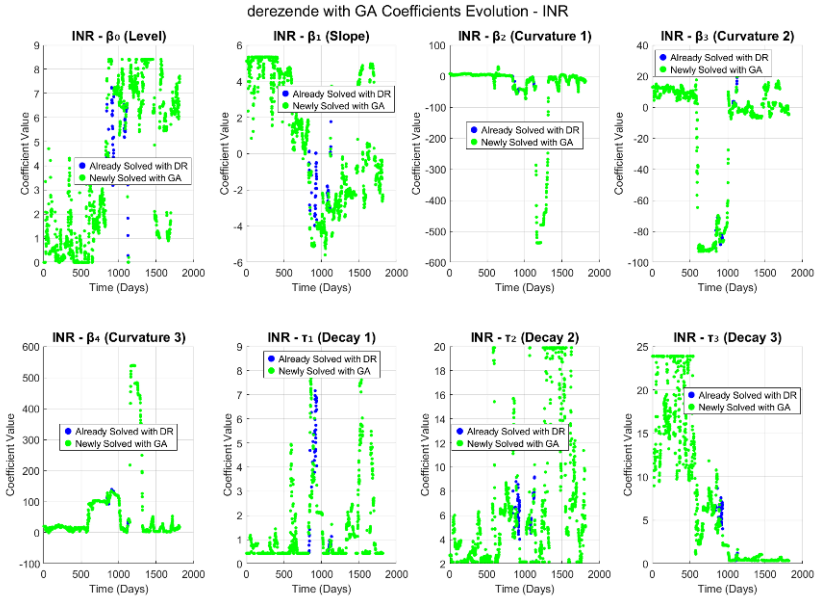


Figure A.137: De Rezende-Ferreira model (GA-optimized) coefficients – INR

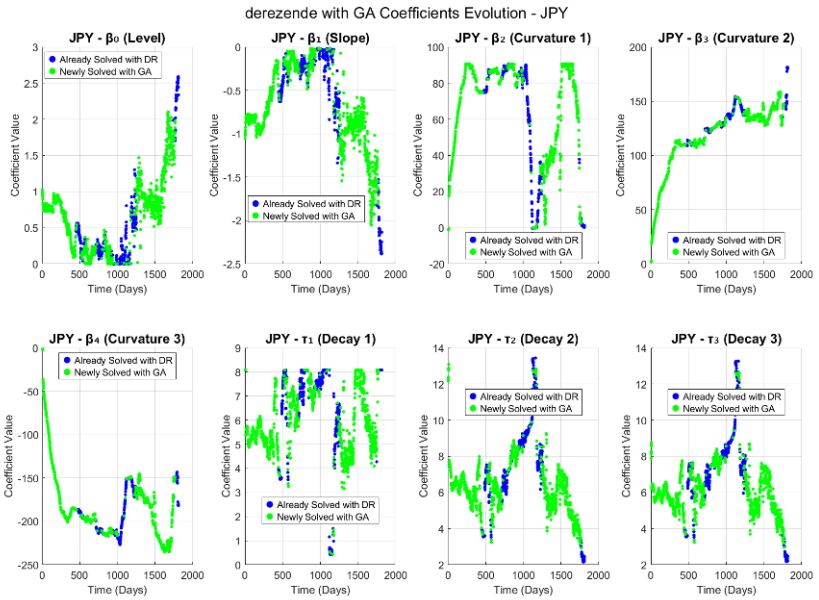


Figure A.138: De Rezende-Ferreira model (GA-optimized) coefficients – JPY

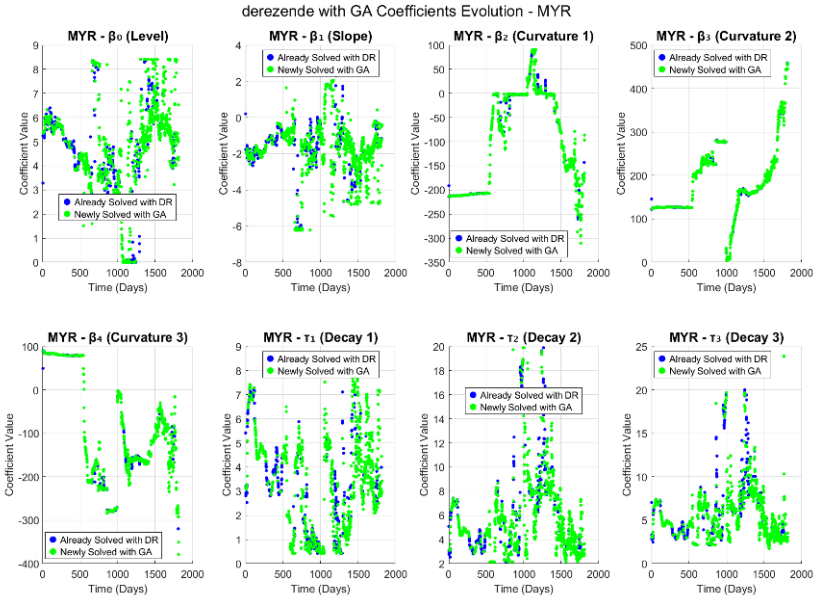


Figure A.139: De Rezende-Ferreira model (GA-optimized) coefficients – MYR

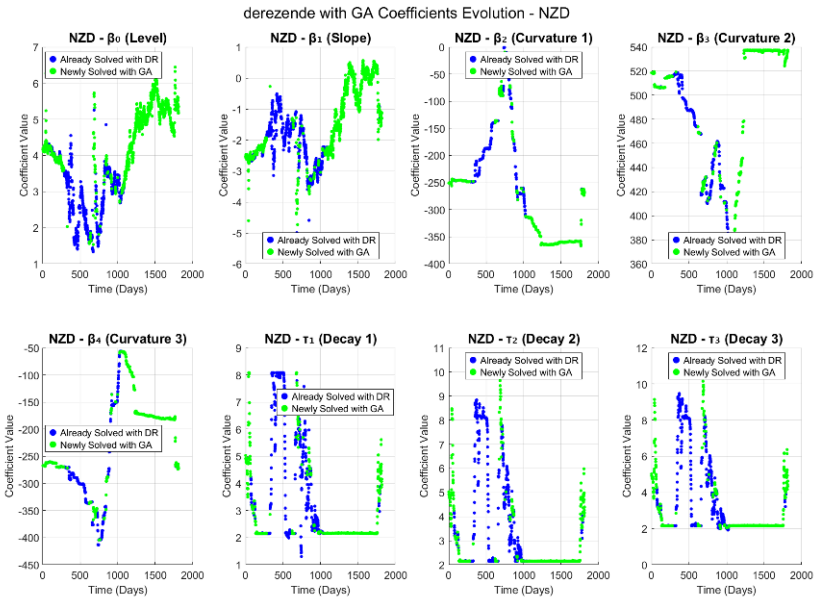


Figure A.140: De Rezende-Ferreira model (GA-optimized) coefficients – NZD

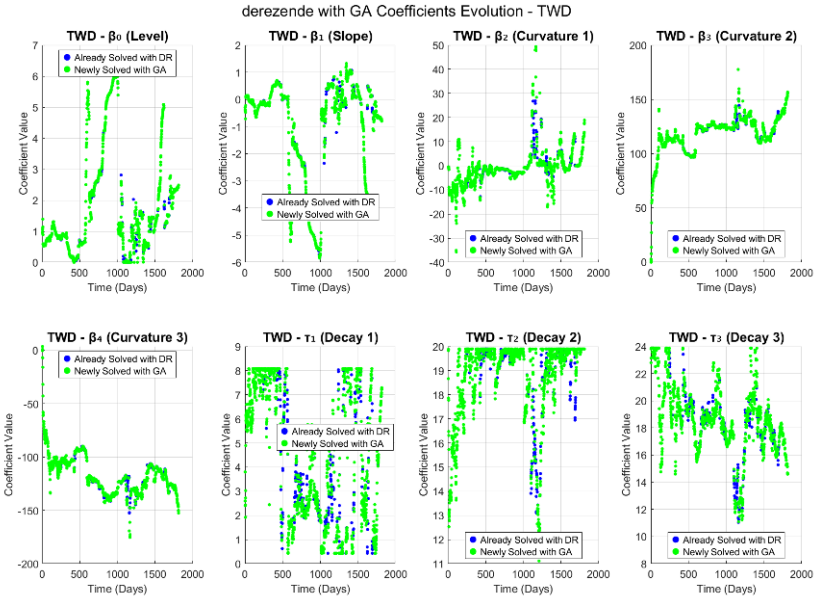


Figure A.141: De Rezende-Ferreira model (GA-optimized) coefficients – TWD

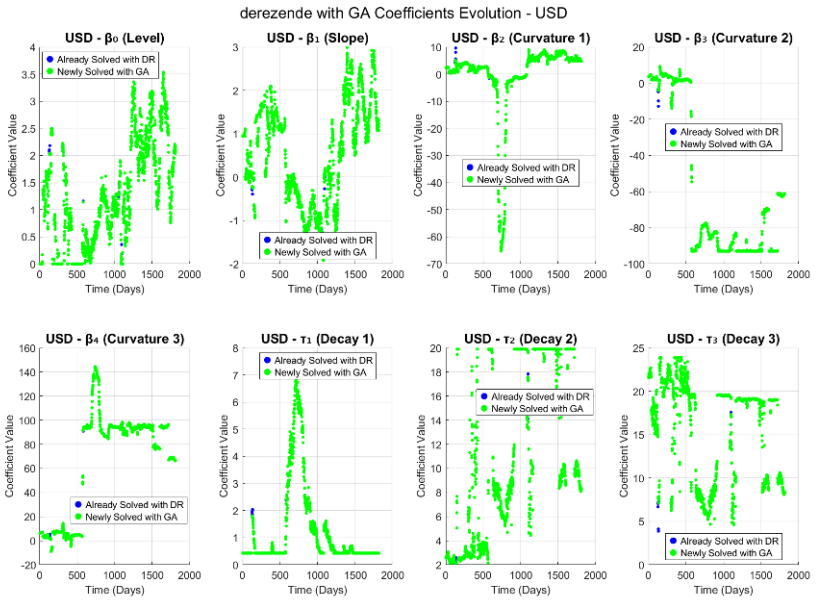


Figure A.142: De Rezende-Ferreira model (GA-optimized) coefficients – USD

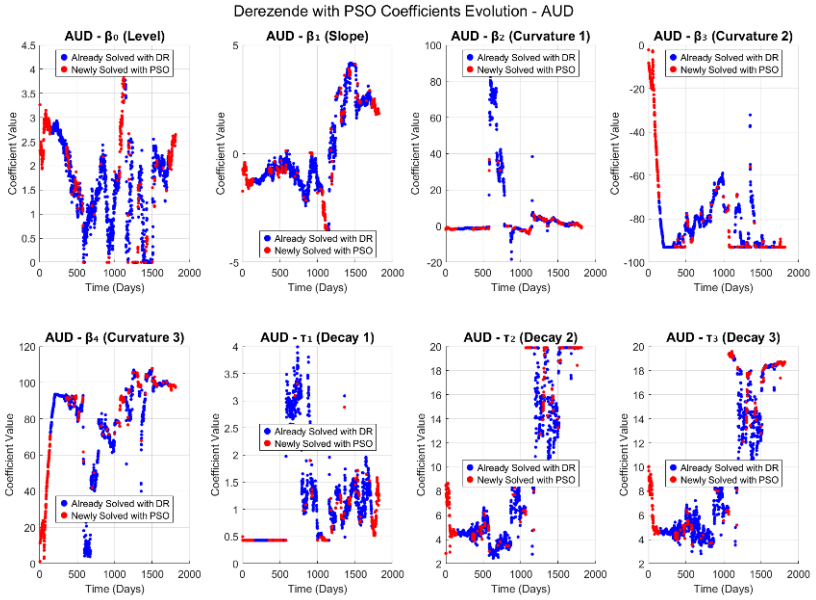


Figure A.143: De Rezende-Ferreira model (PSO-optimized) coefficients – AUD

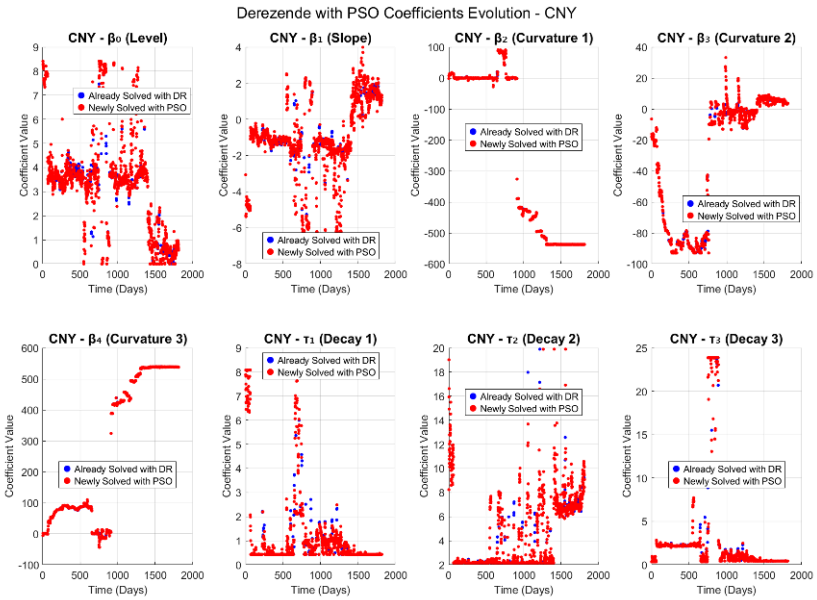


Figure A.144: De Rezende-Ferreira model (PSO-optimized) coefficients – CNY

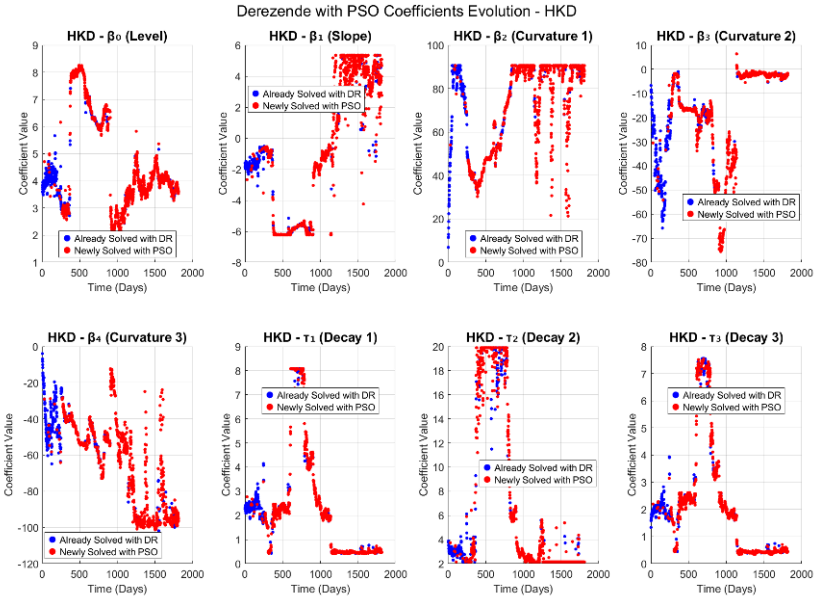


Figure A.145: De Rezende-Ferreira model (PSO-optimized) coefficients – HKD

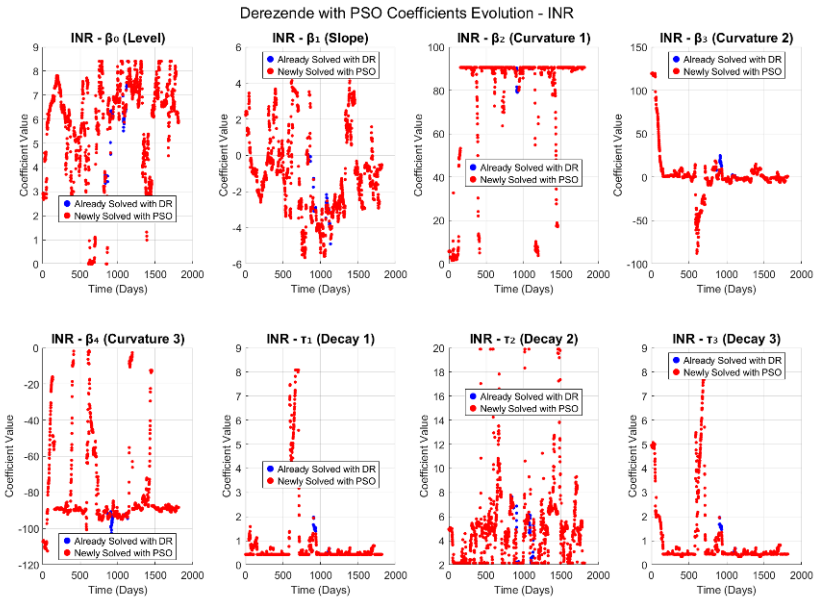


Figure A.146: De Rezende-Ferreira model (PSO-optimized) coefficients – INR

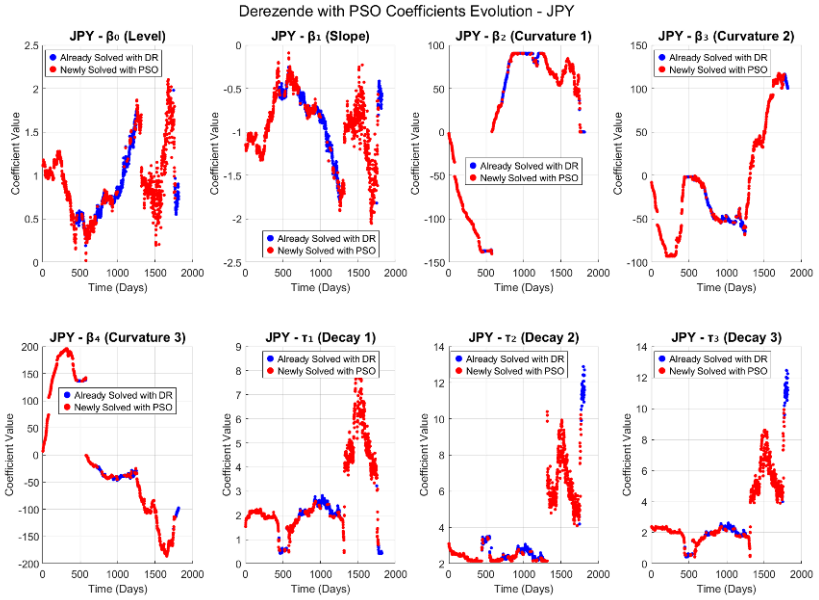


Figure A.147: De Rezende-Ferreira model (PSO-optimized) coefficients – JPY

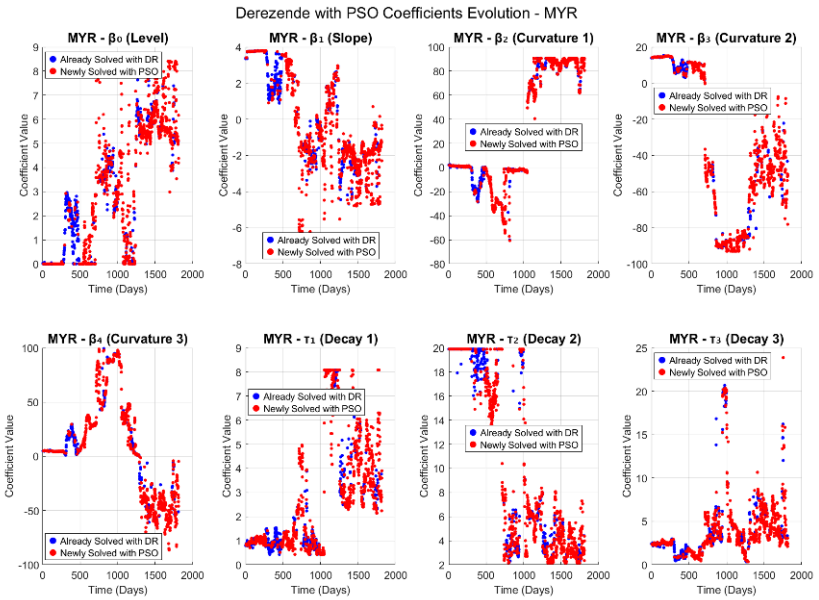


Figure A.148: De Rezende-Ferreira model (PSO-optimized) coefficients – MYR

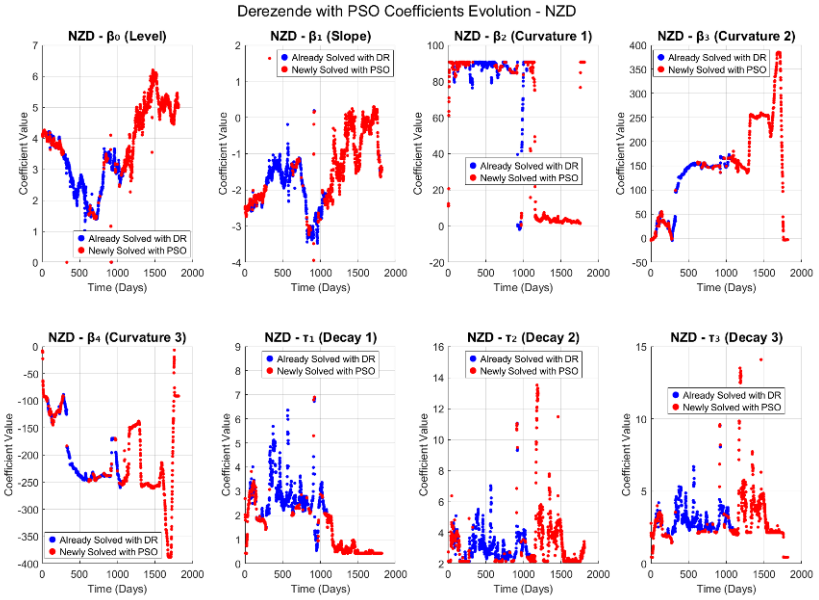


Figure A.149: De Rezende-Ferreira model (PSO-optimized) coefficients – NZD

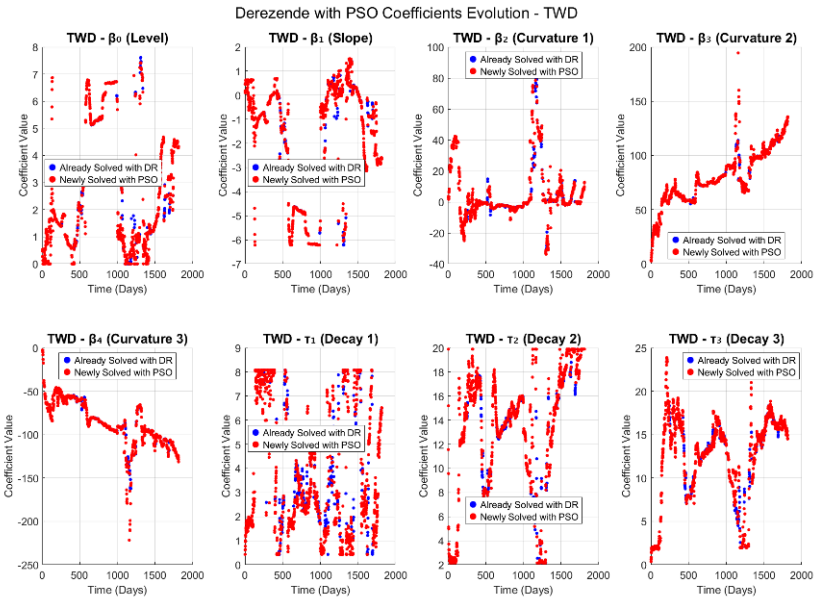


Figure A.150: De Rezende-Ferreira model (PSO-optimized) coefficients – TWD

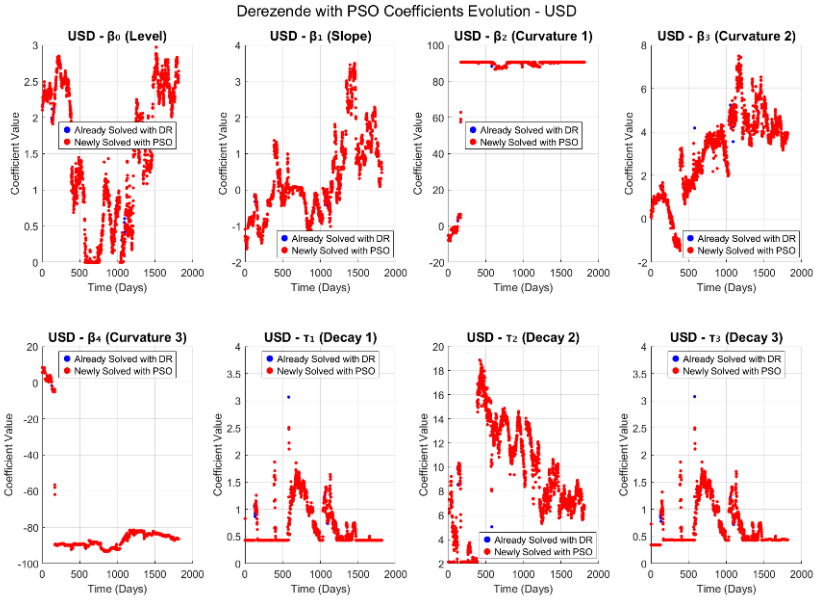


Figure A.151: De Rezende-Ferreira model (PSO-optimized) coefficients – USD

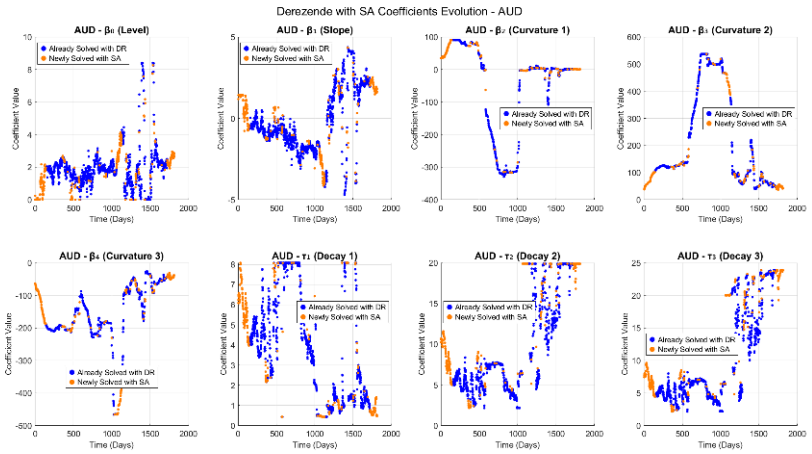


Figure A.152: De Rezende-Ferreira model (SA-optimized) coefficients – AUD

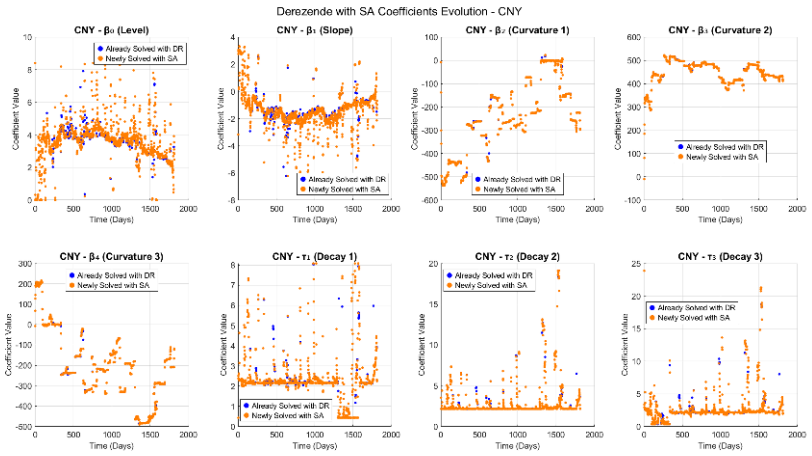


Figure A.153: De Rezende-Ferreira model (SA-optimized) coefficients – CNY

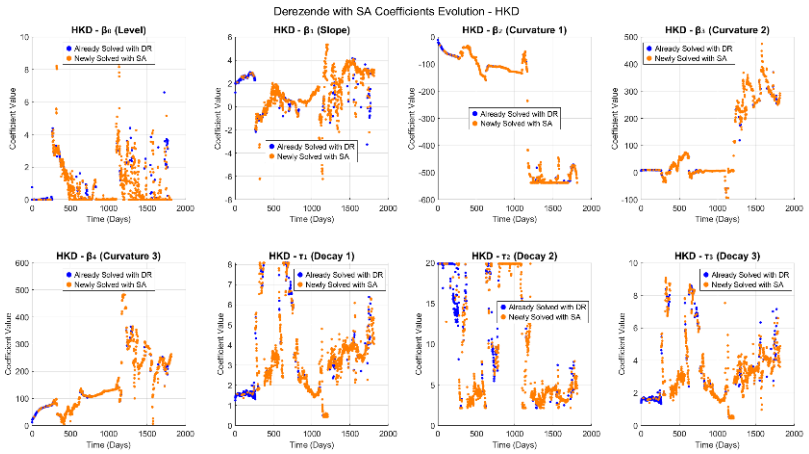


Figure A.154: De Rezende-Ferreira model (SA-optimized) coefficients – HKD

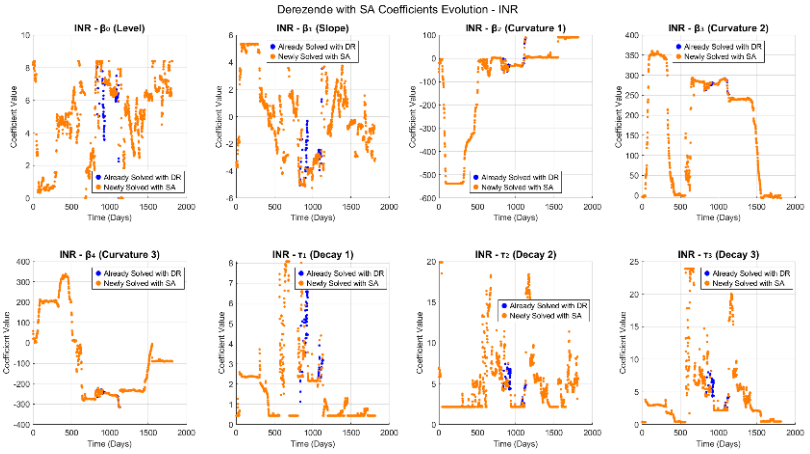


Figure A.155: De Rezende-Ferreira model (SA-optimized) coefficients – INR

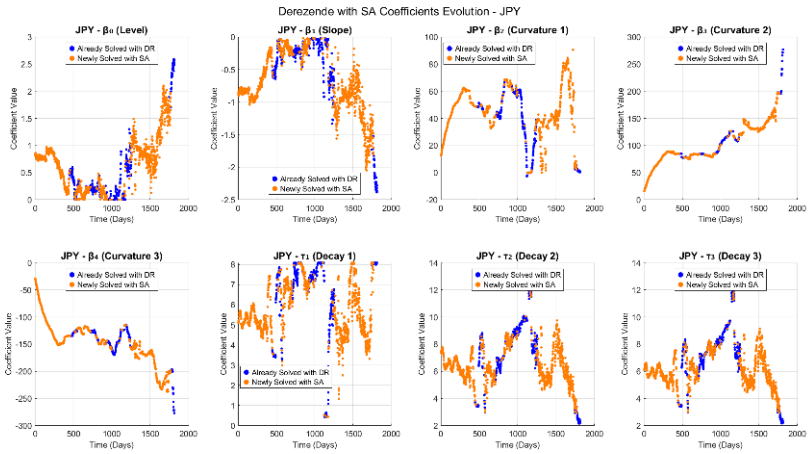


Figure A.156: De Rezende-Ferreira model (SA-optimized) coefficients – JPY

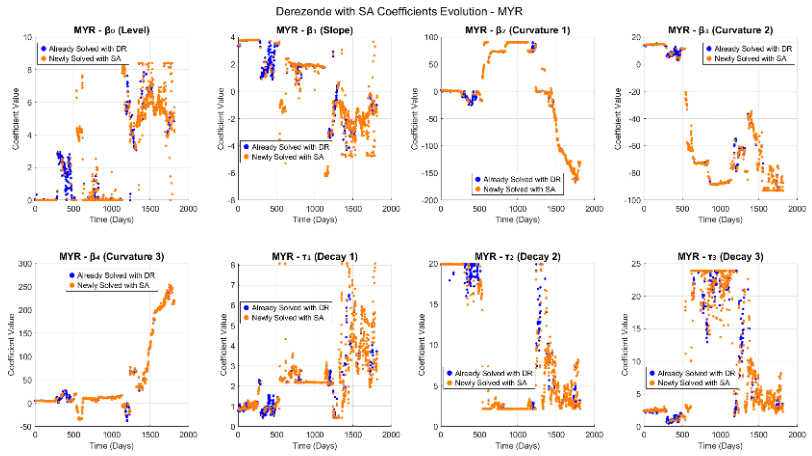


Figure A.157: De Rezende-Ferreira model (SA-optimized) coefficients – MYR

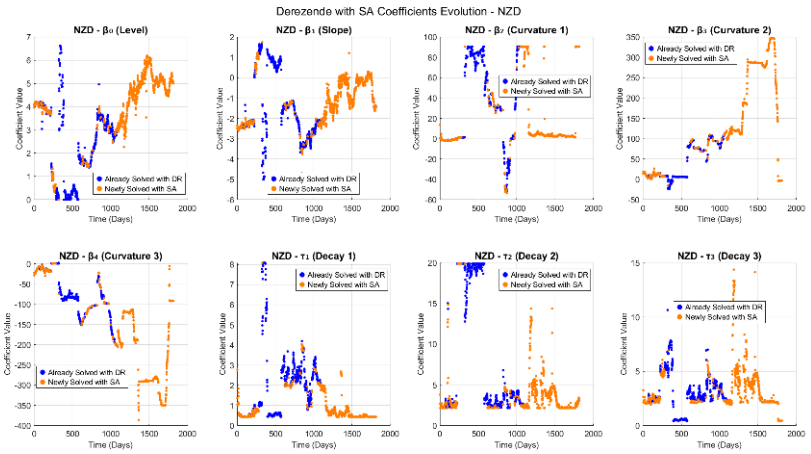


Figure A.158: De Rezende-Ferreira model (SA-optimized) coefficients – NZD

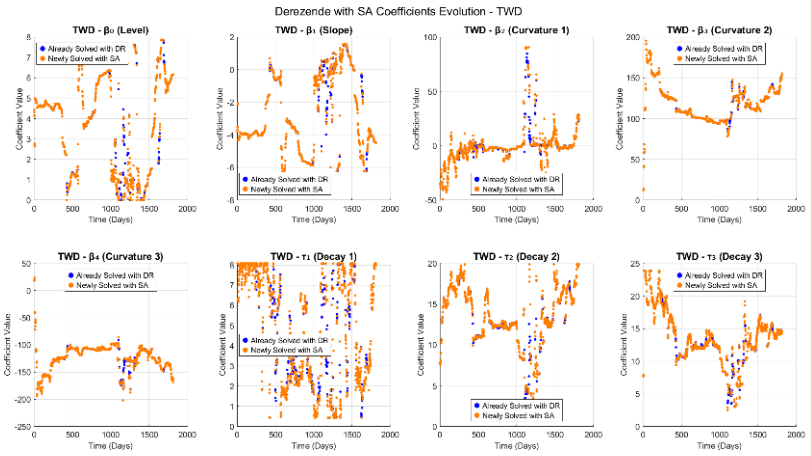


Figure A.159: De Rezende-Ferreira model (SA-optimized) coefficients – TWD

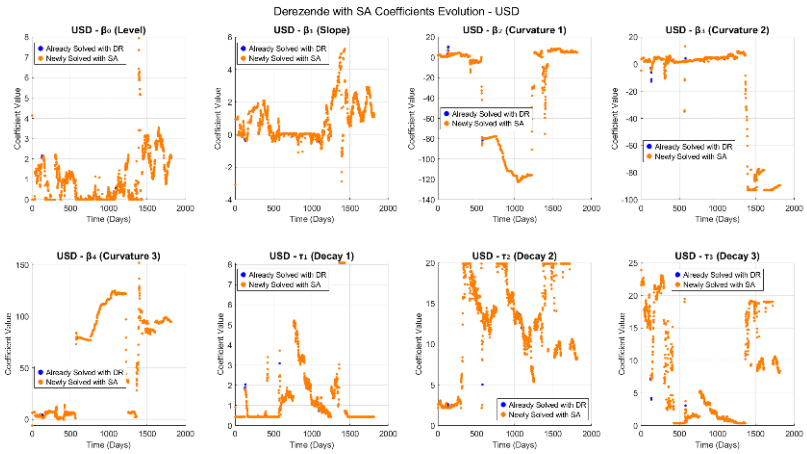


Figure A.160: De Rezende-Ferreira model (SA-optimized) coefficients – USD

A.2.4 Additional Results (Pacific Economies)

To conclude, the following histograms summarize the percentage of yield curves successfully fitted using each method

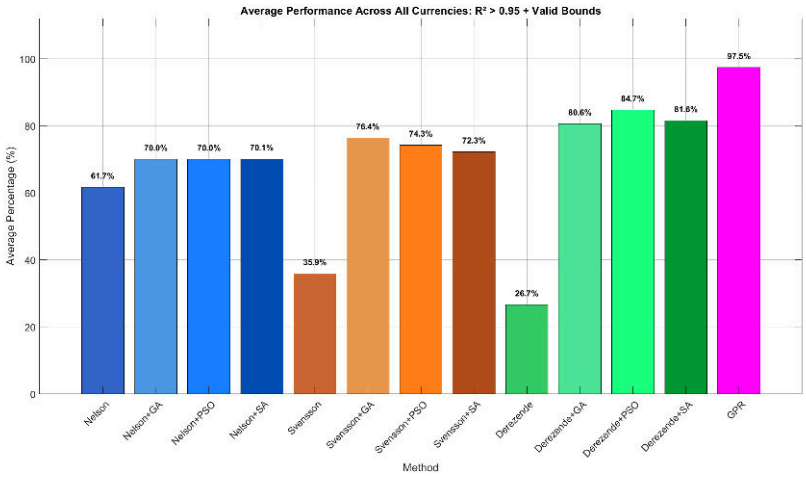


Figure A.161: Overall Average Cumulative R^2 Bounds Comparison

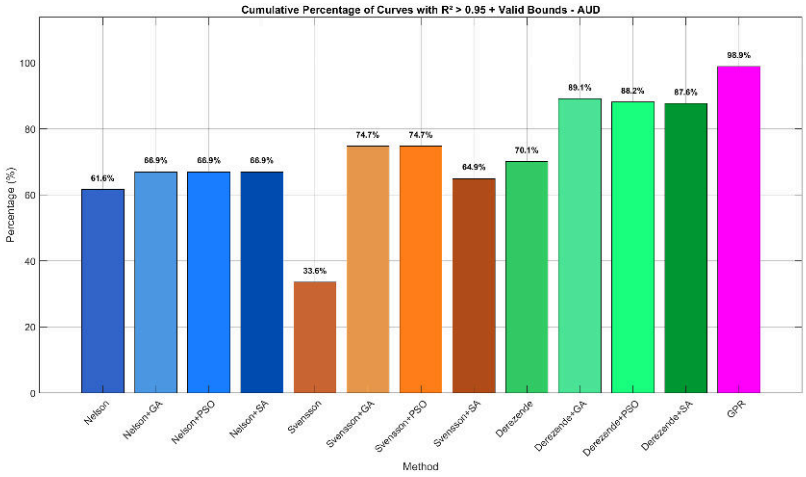


Figure A.162: Cumulative R^2 bounds – AUD

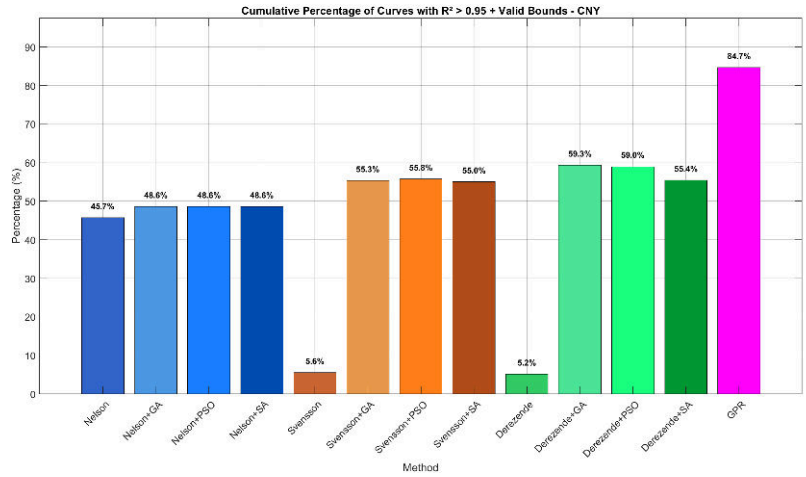


Figure A.163: Cumulative R^2 bounds – CNY

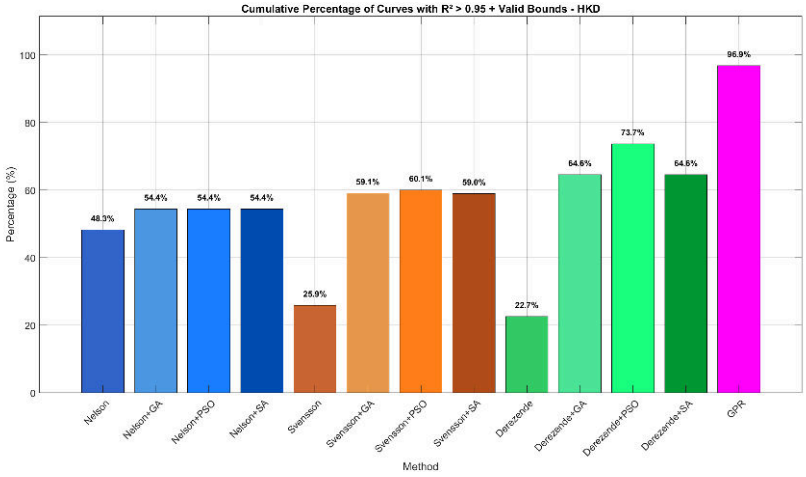


Figure A.164: Cumulative R^2 bounds – HKD

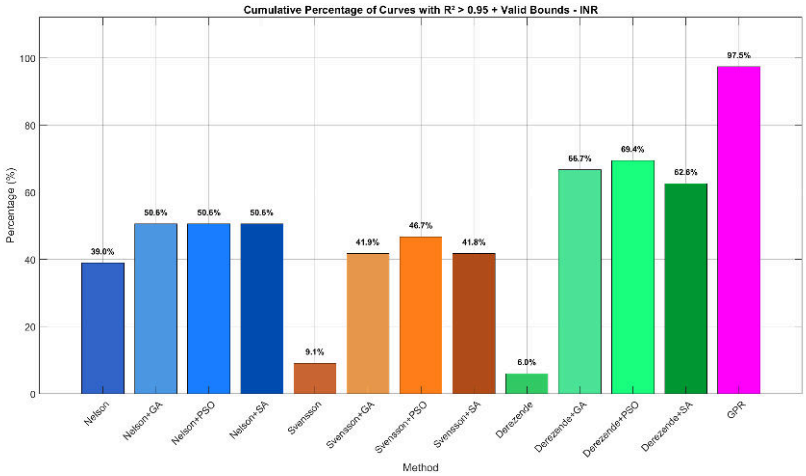


Figure A.165: Cumulative R^2 bounds – INR

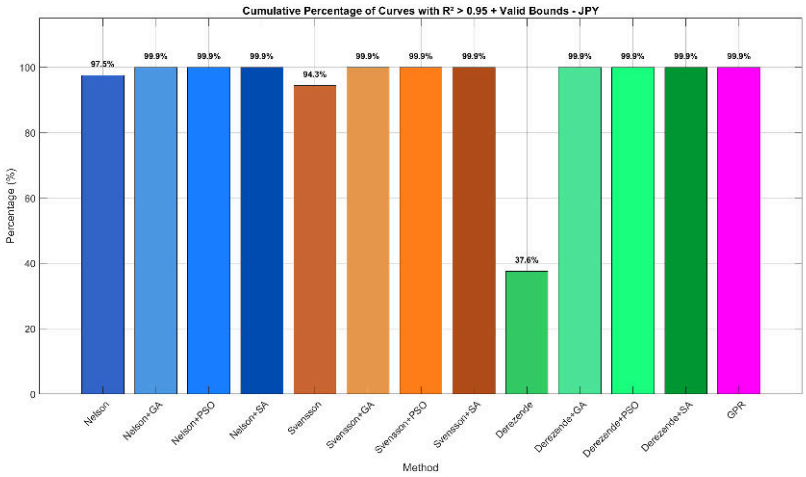


Figure A.166: Cumulative R^2 bounds – JPY

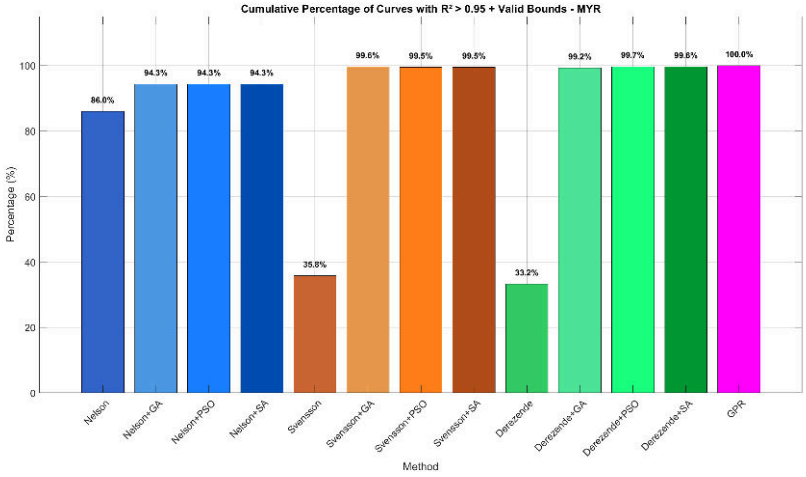


Figure A.167: Cumulative R^2 bounds – MYR

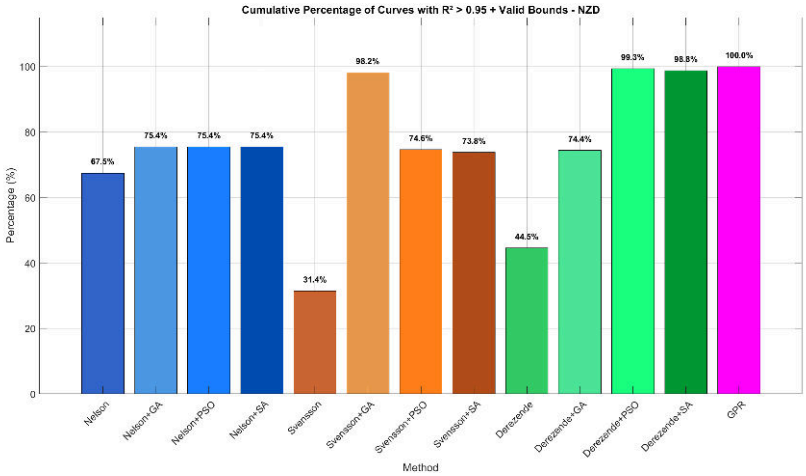


Figure A.168: Cumulative R^2 bounds – NZD

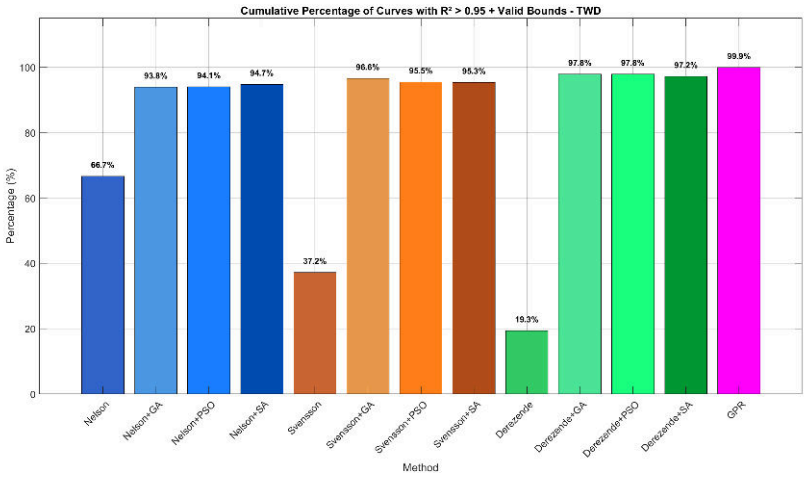


Figure A.169: Cumulative R^2 bounds – TWD

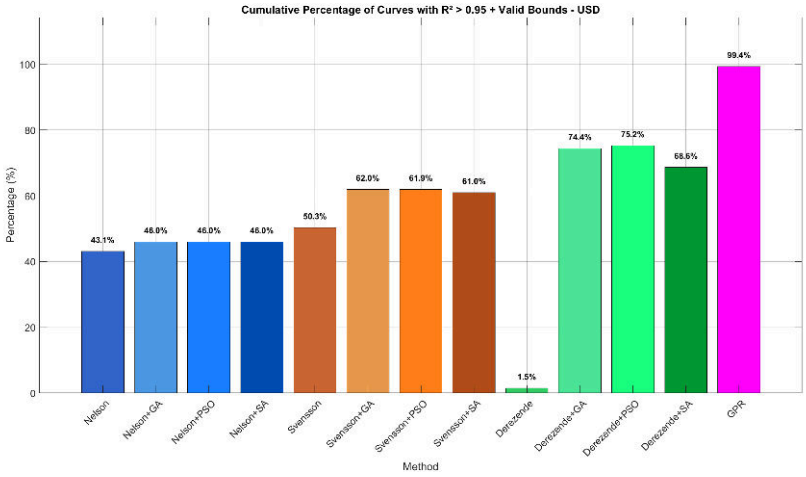


Figure A.170: Cumulative R^2 bounds – USD

A.3 Sovereign Yield Curves (European Countries)

A.3.1 Performance Comparison (Base vs. Optimized)

The following figures present the performance comparison between the baseline fitting (shown in black) and the optimized versions obtained via three distinct algorithms: genetic algorithm (GA, in green), particle swarm optimization (PSO, in orange), and simulated annealing (SA, in red)

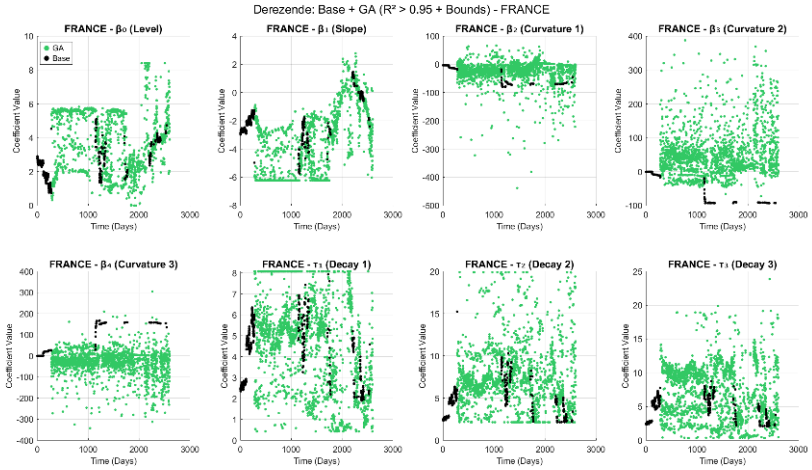


Figure A.171: De Rezende-Ferreira Model: Base vs. GA Optimized Performance – FRANCE

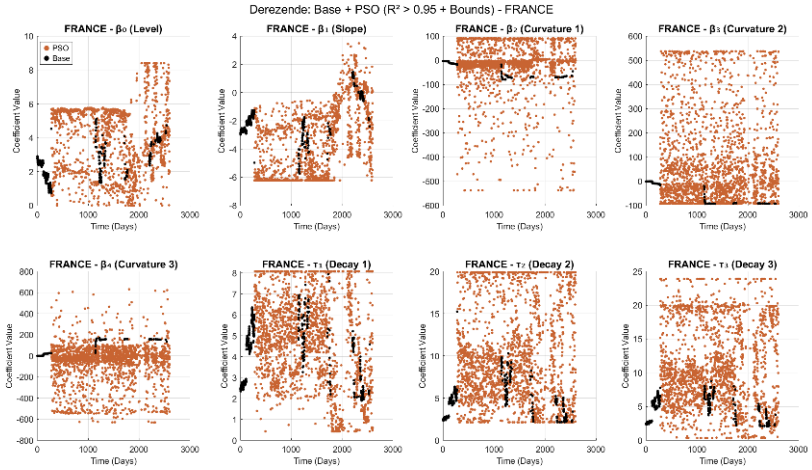


Figure A.172: De Rezende-Ferreira Model: Base vs. PSO Optimized Performance – FRANCE

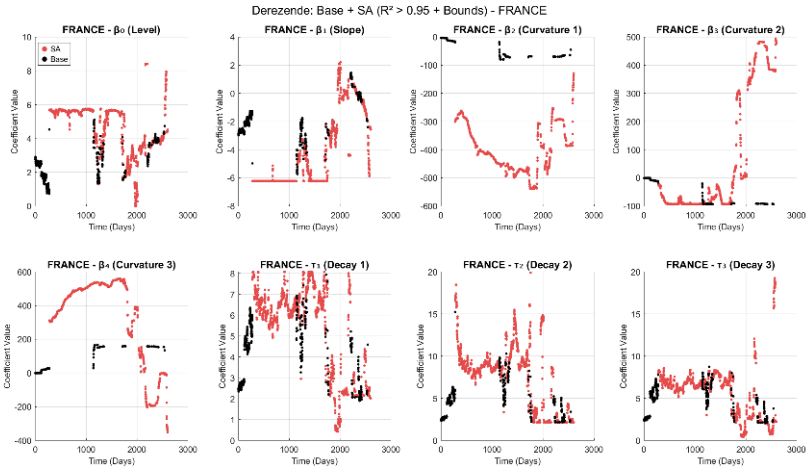


Figure A.173: De Rezende-Ferreira Model: Base vs. SA Optimized Performance – FRANCE

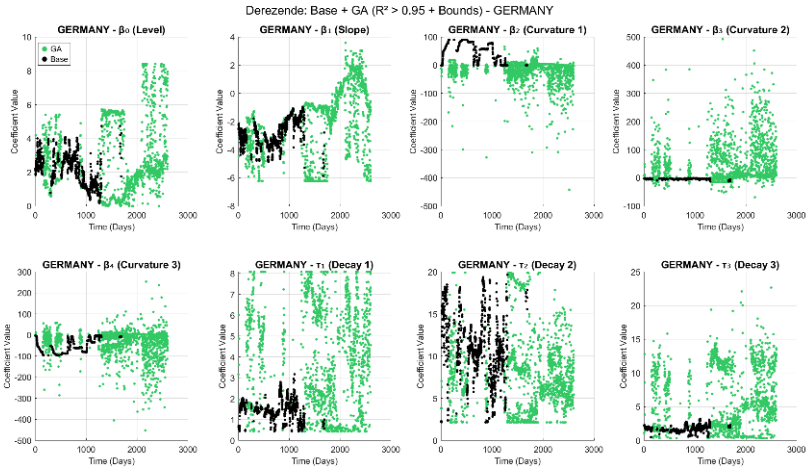


Figure A.174: De Rezende-Ferreira Model: Base vs. GA Optimized Performance – GERMANY

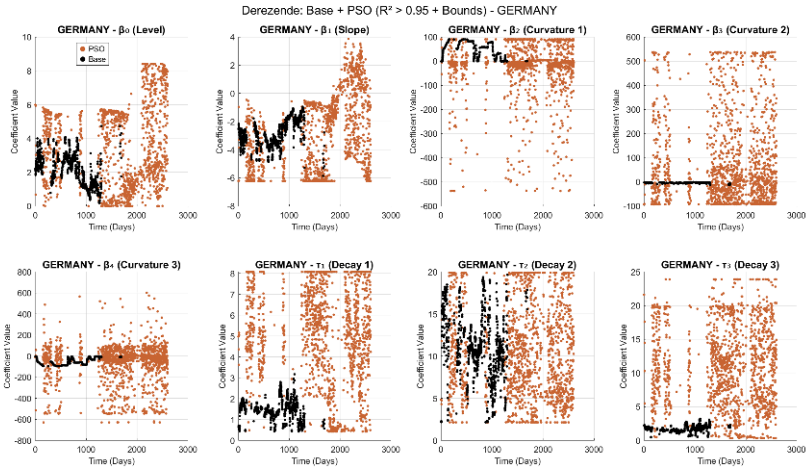


Figure A.175: De Rezende-Ferreira Model: Base vs. PSO Optimized Performance – GERMANY

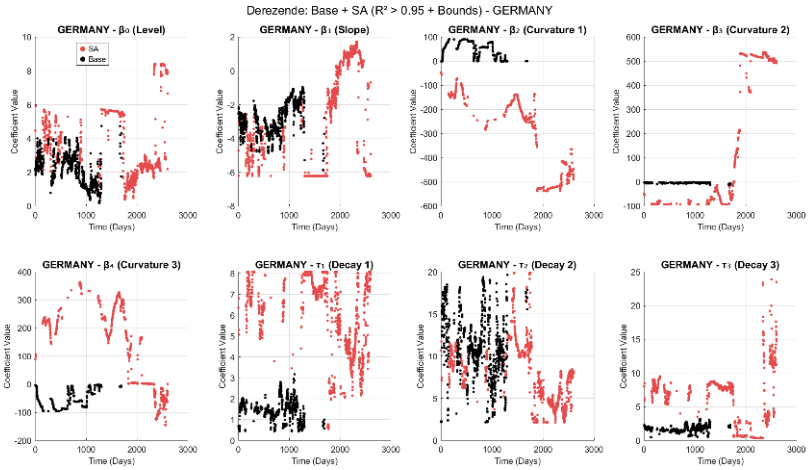


Figure A.176: De Rezende-Ferreira Model: Base vs. SA Optimized Performance – GERMANY

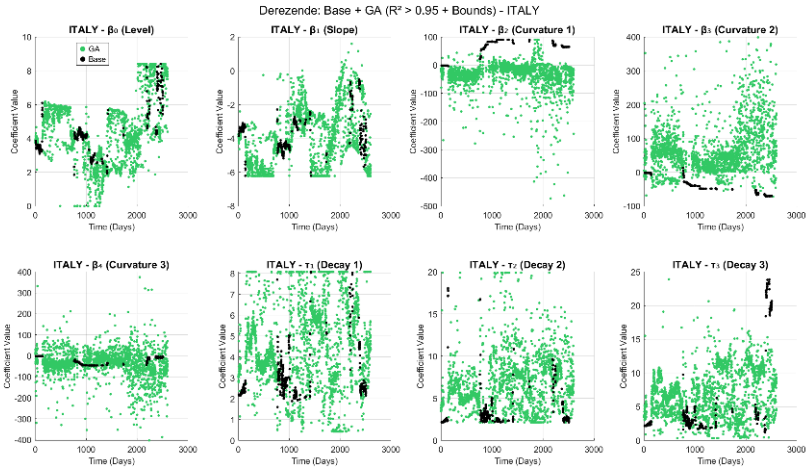


Figure A.177: De Rezende-Ferreira Model: Base vs. GA Optimized Performance – ITALY

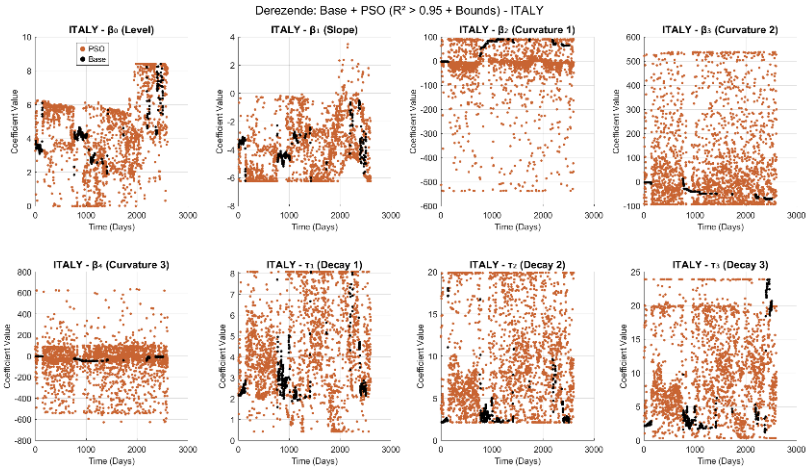


Figure A.178: De Rezende-Ferreira Model: Base vs. PSO Optimized Performance – ITALY

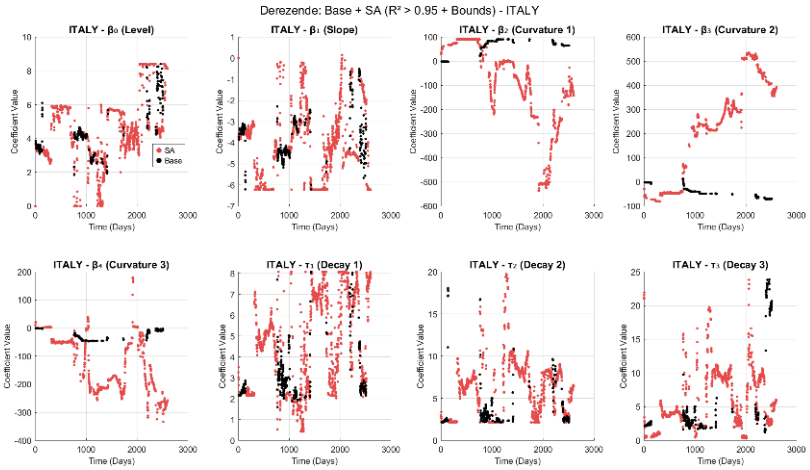


Figure A.179: De Rezende-Ferreira Model: Base vs. SA Optimized Performance – ITALY

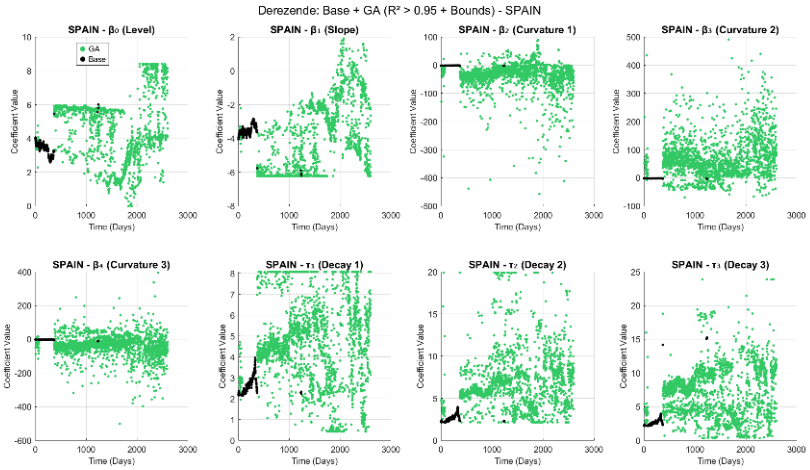


Figure A.180: De Rezende-Ferreira Model: Base vs. GA Optimized Performance – SPAIN

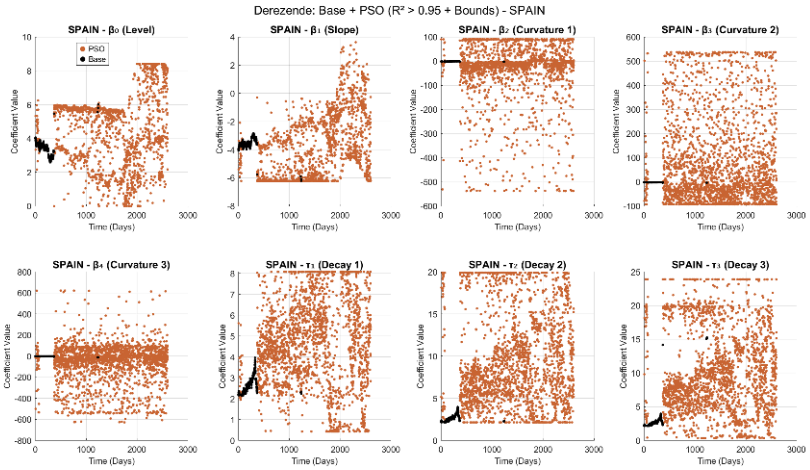


Figure A.181: De Rezende-Ferreira Model: Base vs. PSO Optimized Performance – SPAIN

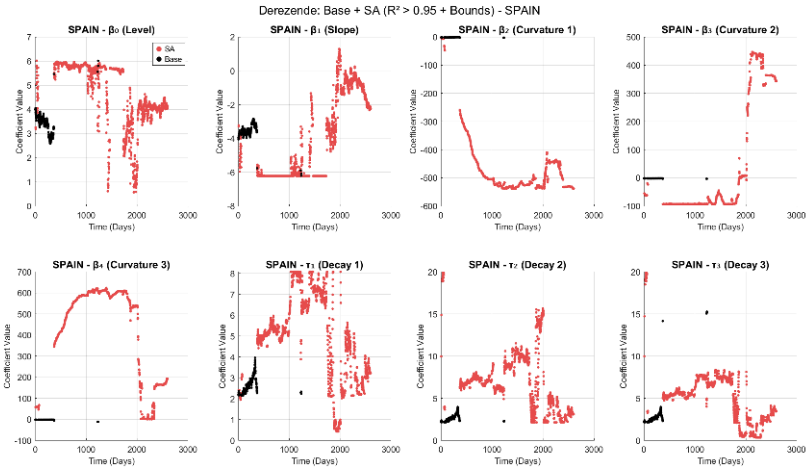


Figure A.182: De Rezende-Ferreira Model: Base vs. SA Optimized Performance – SPAIN

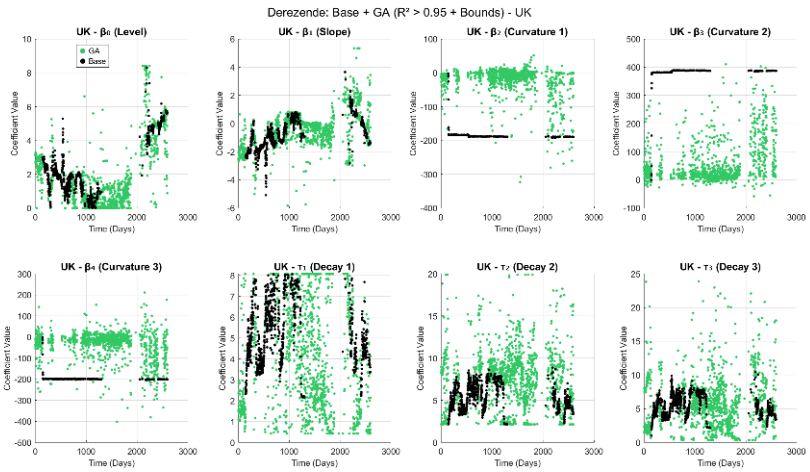


Figure A.183: De Rezende-Ferreira Model: Base vs. GA Optimized Performance – UK

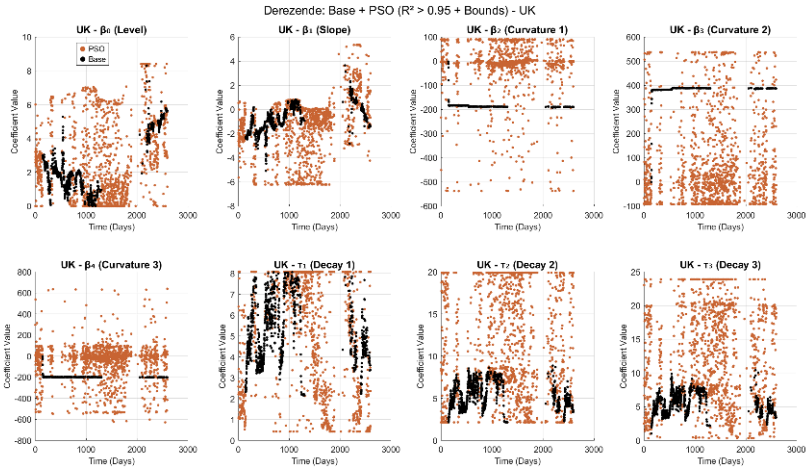


Figure A.184: De Rezende-Ferreira Model: Base vs. PSO Optimized Performance – UK

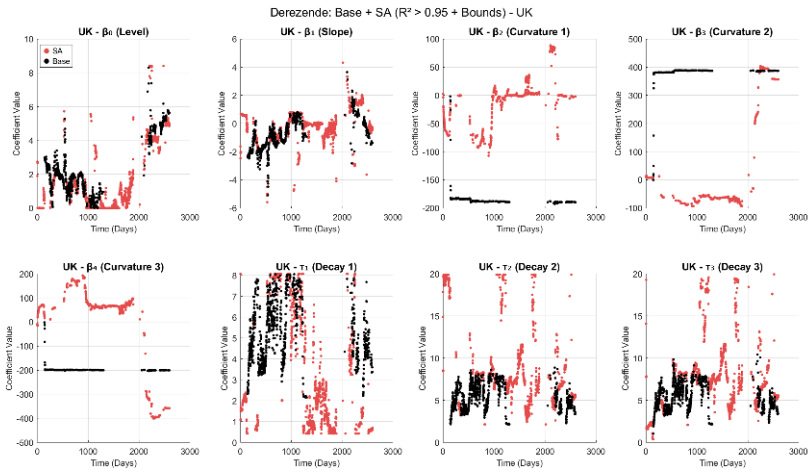


Figure A.185: De Rezende-Ferreira Model: Base vs. SA Optimized Performance – UK

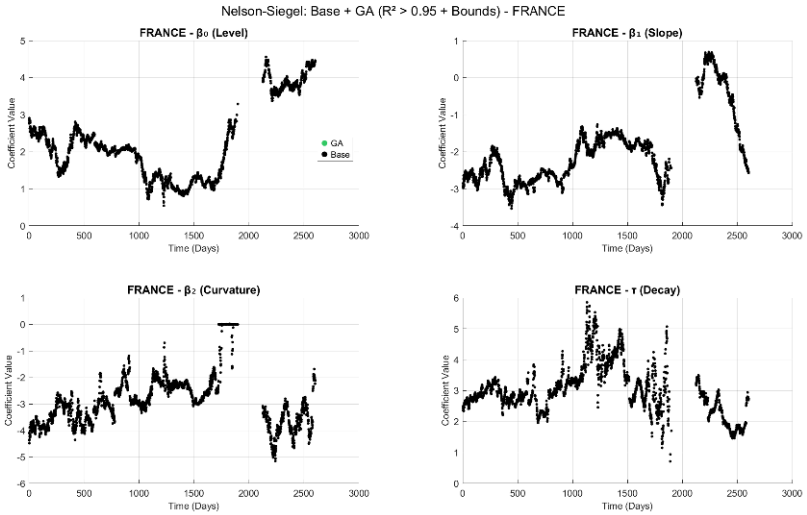


Figure A.186: Nelson-Siegel Model: Base vs. GA Optimized Performance – FRANCE

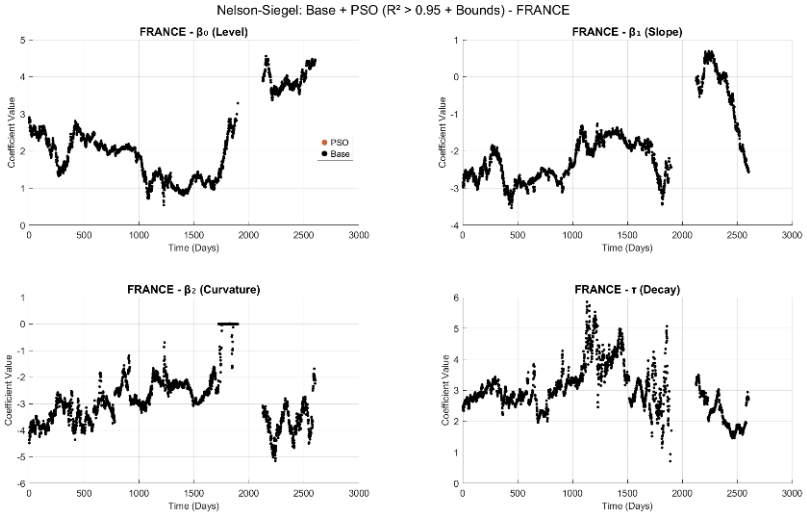


Figure A.187: Nelson-Siegel Model: Base vs. PSO Optimized Performance – FRANCE

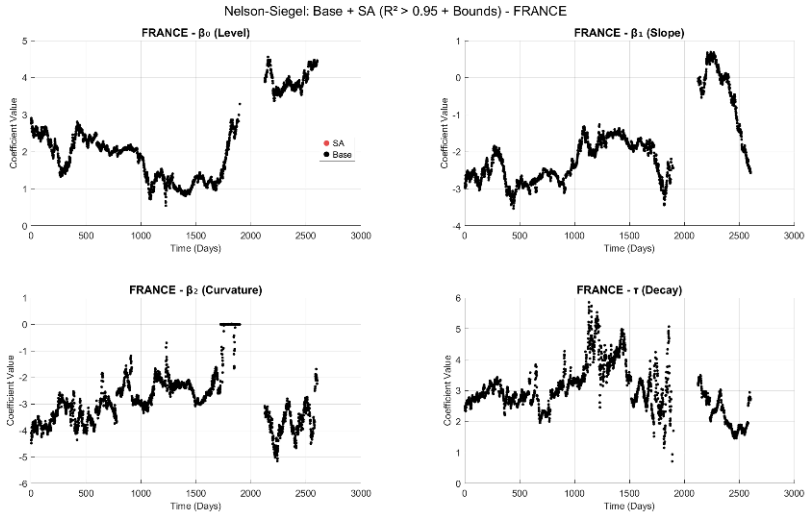


Figure A.188: Nelson-Siegel Model: Base vs. SA Optimized Performance – FRANCE

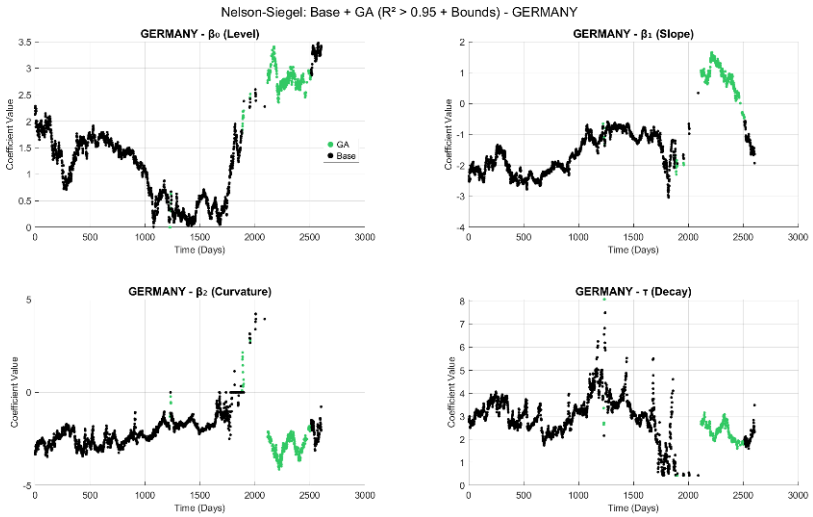


Figure A.189: Nelson-Siegel Model: Base vs. GA Optimized Performance – GERMANY

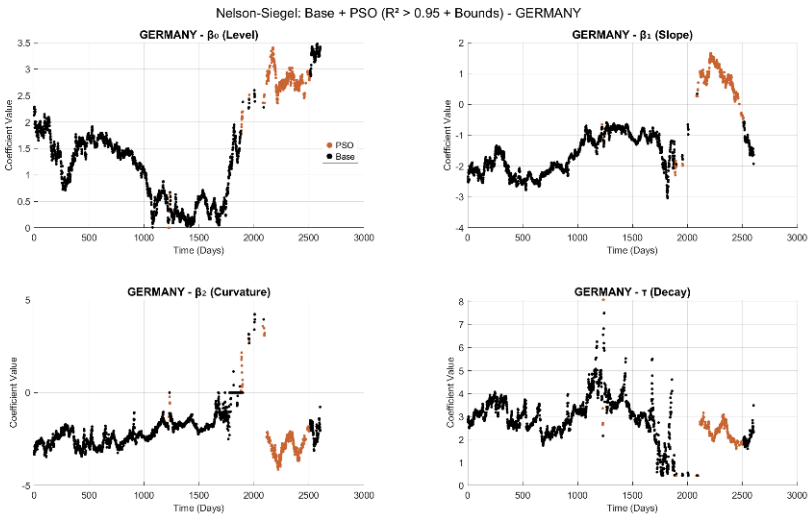


Figure A.190: Nelson-Siegel Model: Base vs. PSO Optimized Performance – GERMANY

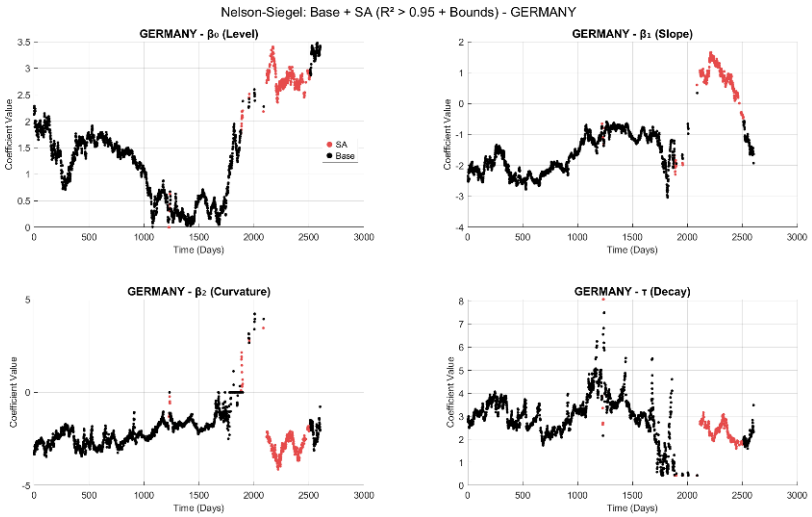


Figure A.191: Nelson-Siegel Model: Base vs. SA Optimized Performance – GERMANY

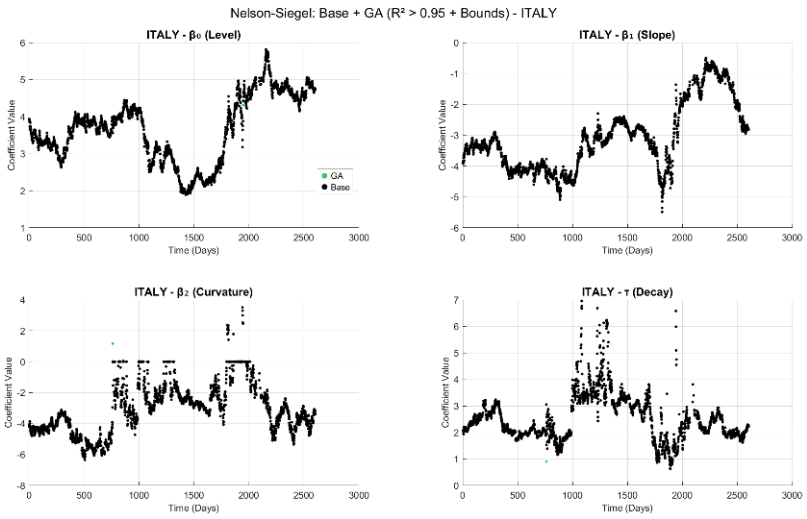


Figure A.192: Nelson-Siegel Model: Base vs. GA Optimized Performance – ITALY

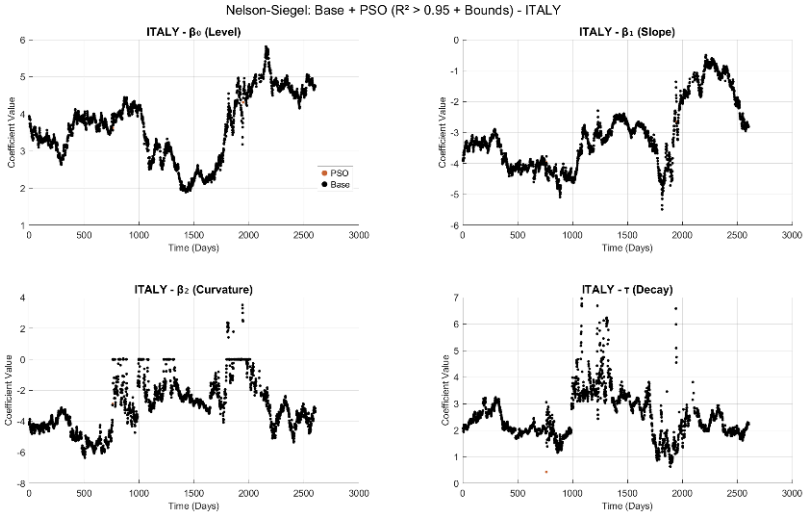


Figure A.193: Nelson-Siegel Model: Base vs. PSO Optimized Performance – ITALY

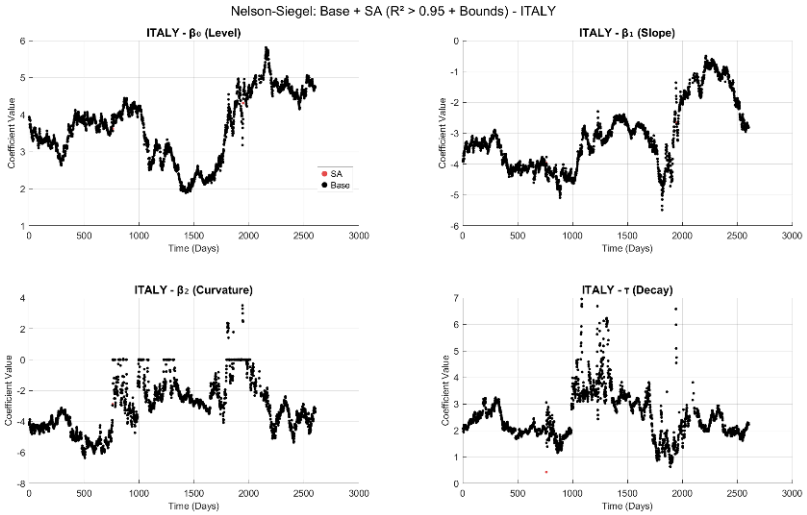


Figure A.194: Nelson-Siegel Model: Base vs. SA Optimized Performance – ITALY

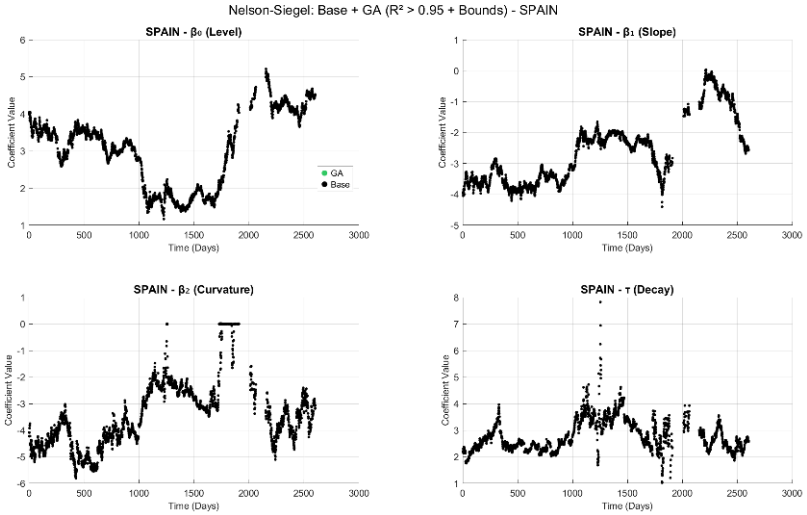


Figure A.195: Nelson-Siegel Model: Base vs. GA Optimized Performance – SPAIN

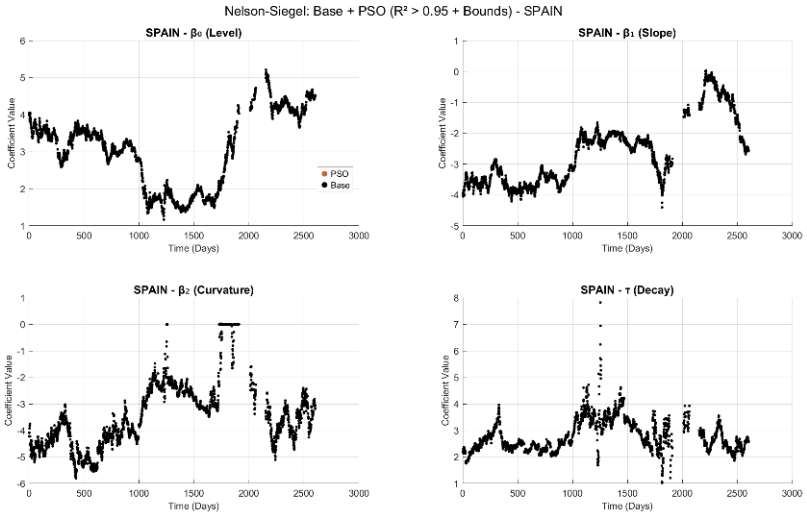


Figure A.196: Nelson-Siegel Model: Base vs. PSO Optimized Performance – SPAIN

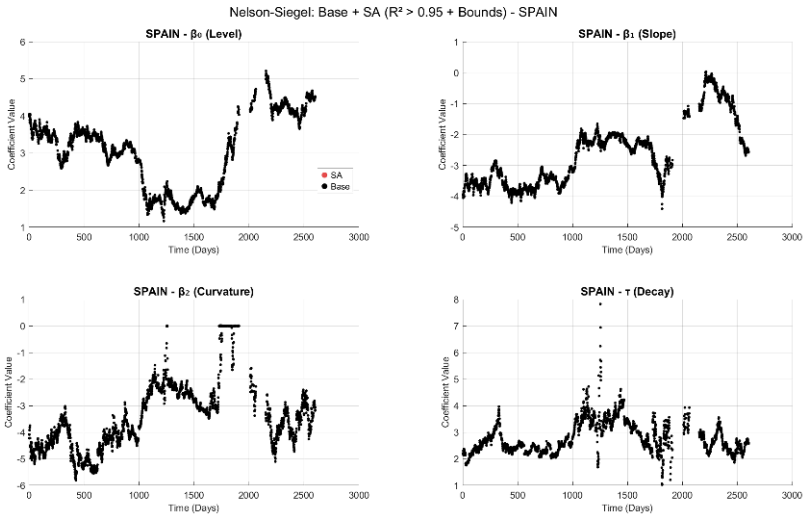


Figure A.197: Nelson-Siegel Model: Base vs. SA Optimized Performance – SPAIN

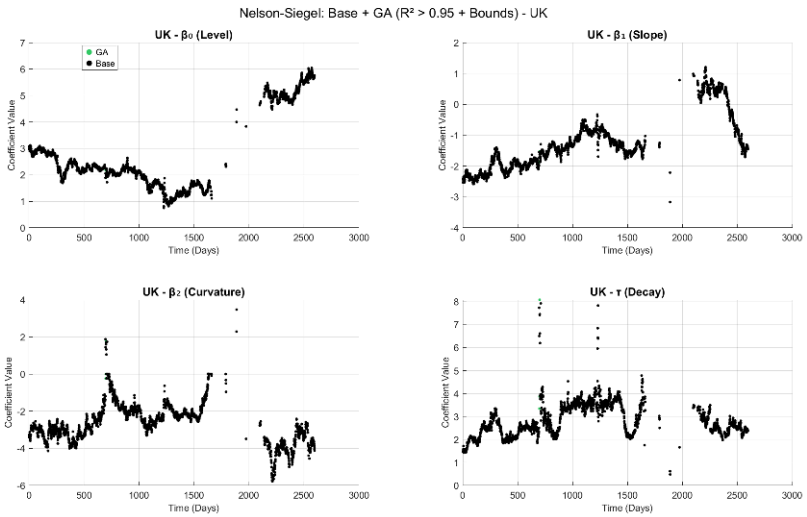


Figure A.198: Nelson-Siegel Model: Base vs. GA Optimized Performance – UK

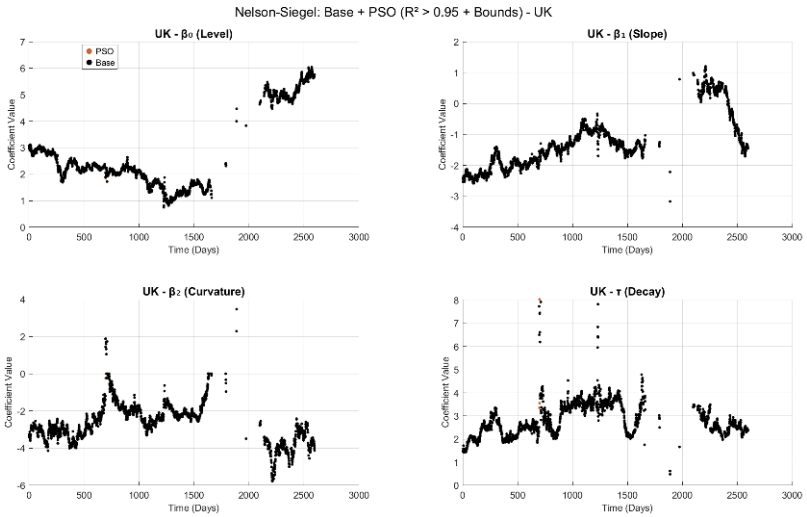


Figure A.199: Nelson-Siegel Model: Base vs. PSO Optimized Performance – UK

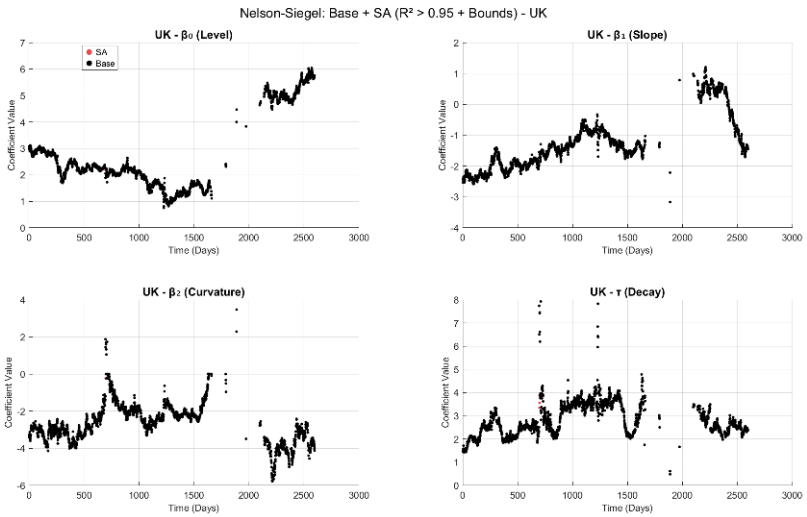


Figure A.200: Nelson-Siegel Model: Base vs. SA Optimized Performance – UK

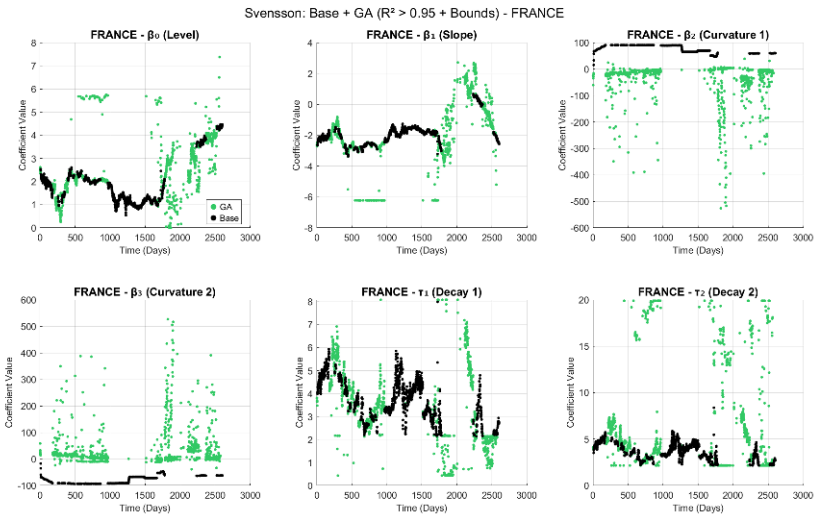


Figure A.201: Svensson Model: Base vs. GA Optimized Performance – FRANCE

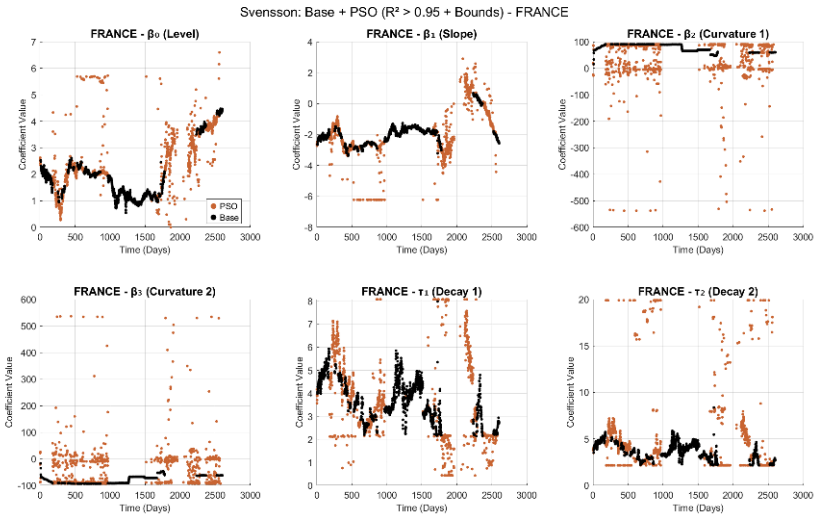


Figure A.202: Svensson Model: Base vs. PSO Optimized Performance – FRANCE

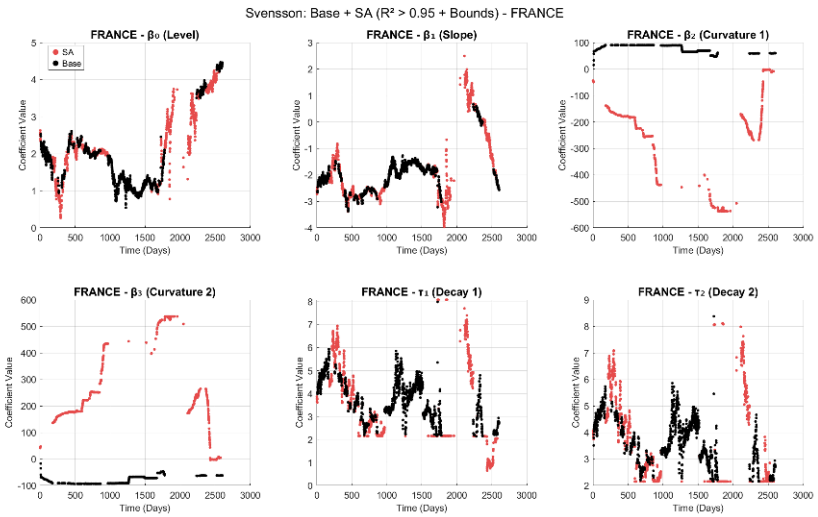


Figure A.203: Svensson Model: Base vs. SA Optimized Performance – FRANCE

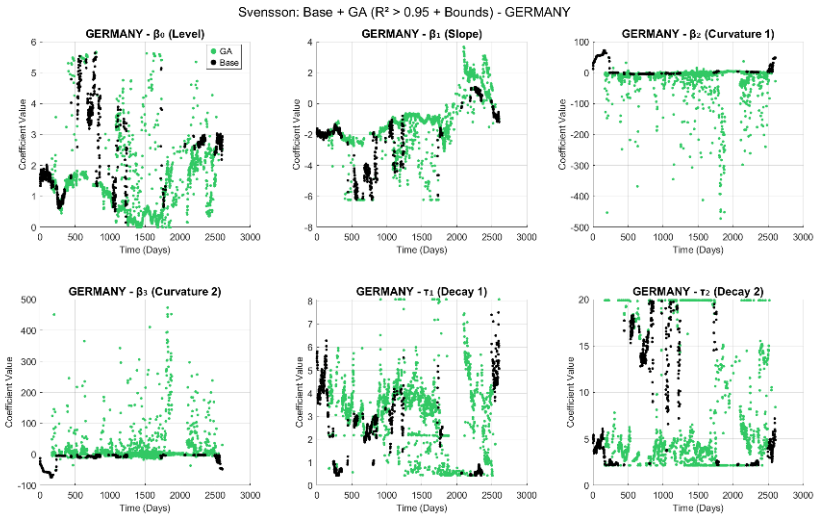


Figure A.204: Svensson Model: Base vs. GA Optimized Performance – GERMANY

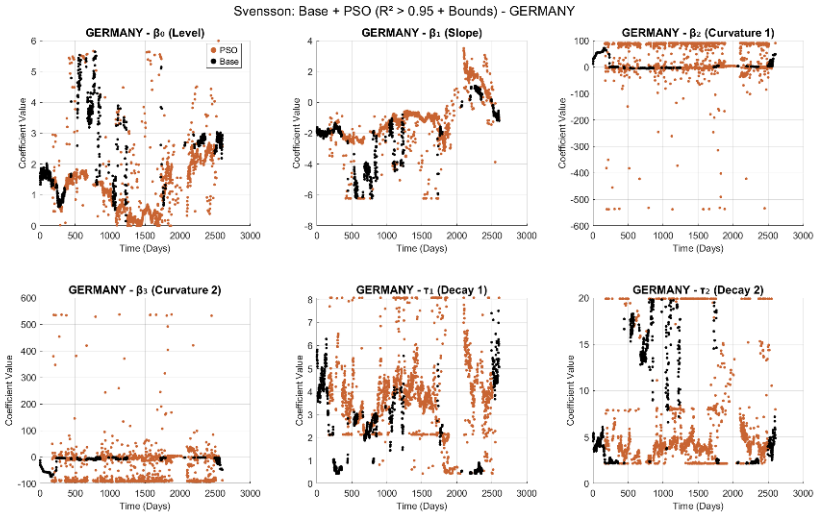


Figure A.205: Svensson Model: Base vs. PSO Optimized Performance – GERMANY

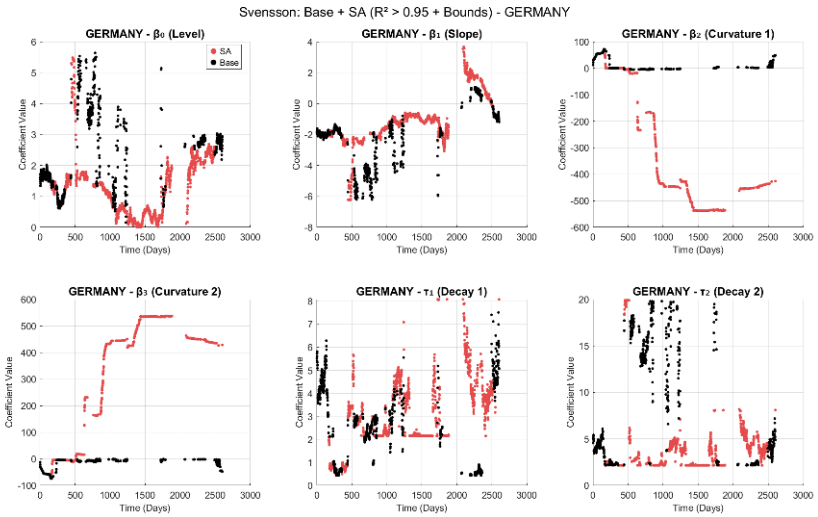


Figure A.206: Svensson Model: Base vs. SA Optimized Performance – GERMANY

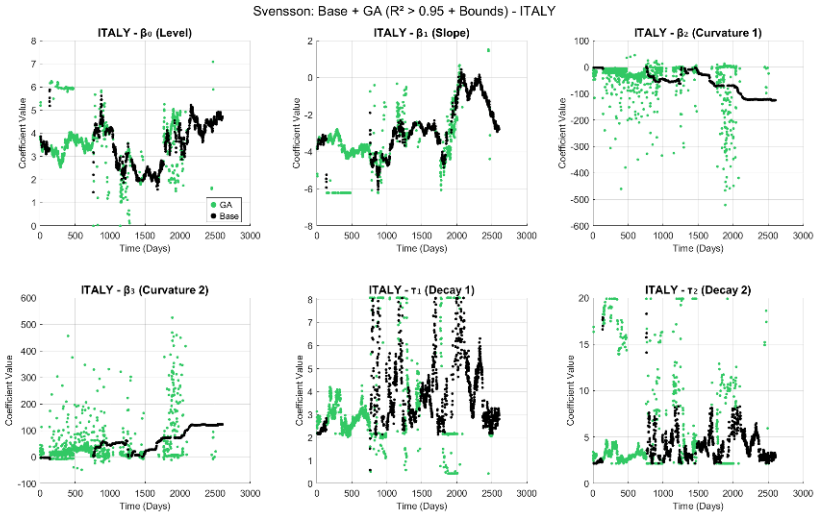


Figure A.207: Svensson Model: Base vs. GA Optimized Performance – ITALY

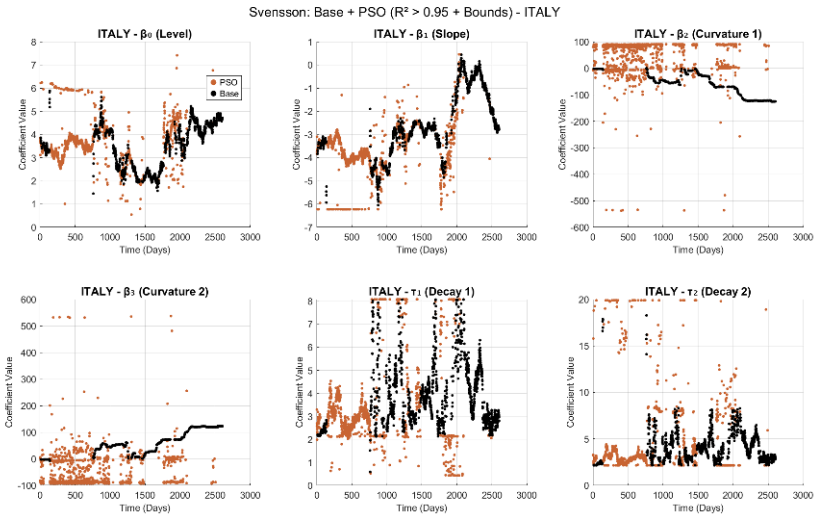


Figure A.208: Svensson Model: Base vs. PSO Optimized Performance – ITALY

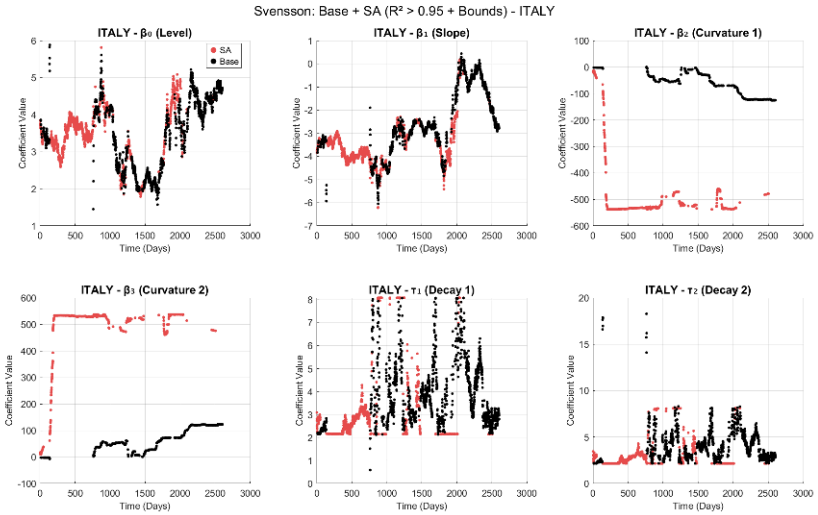


Figure A.209: Svensson Model: Base vs. SA Optimized Performance – ITALY

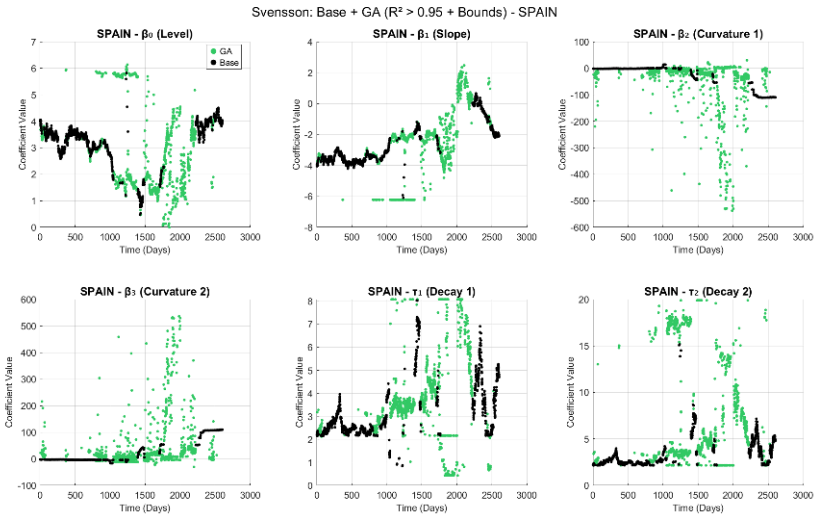


Figure A.210: Svensson Model: Base vs. GA Optimized Performance – SPAIN

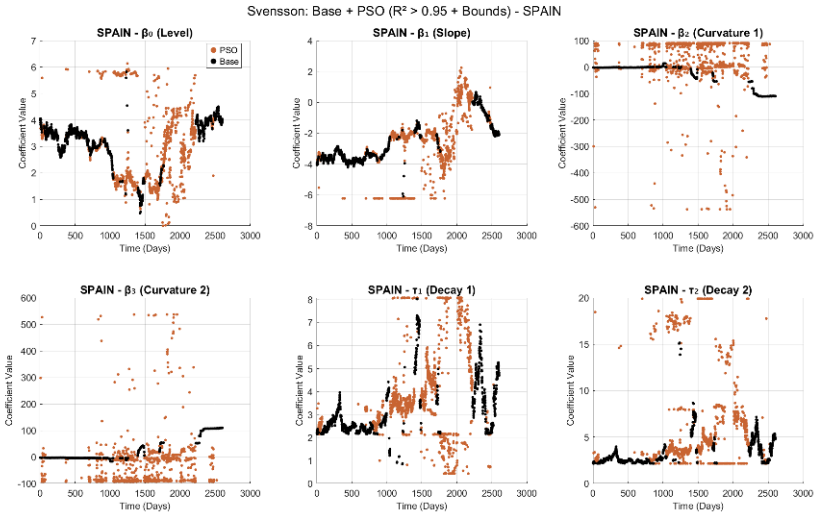


Figure A.211: Svensson Model: Base vs. PSO Optimized Performance – SPAIN

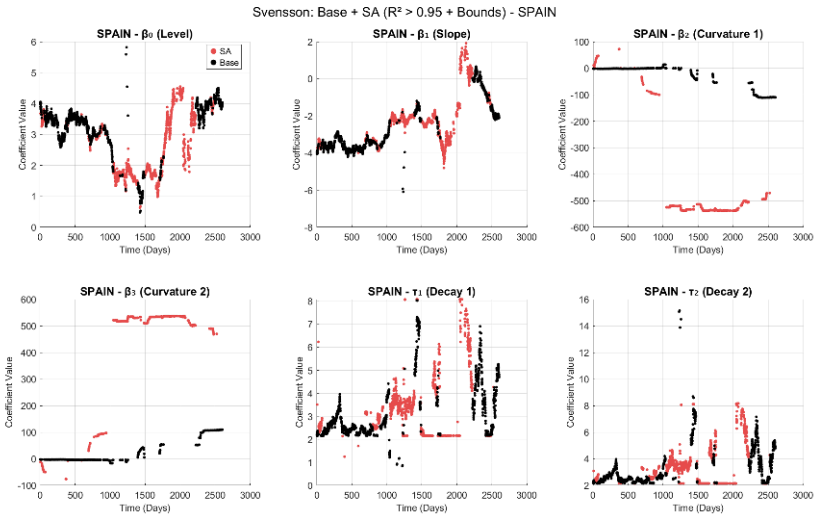


Figure A.212: Svensson Model: Base vs. SA Optimized Performance – SPAIN

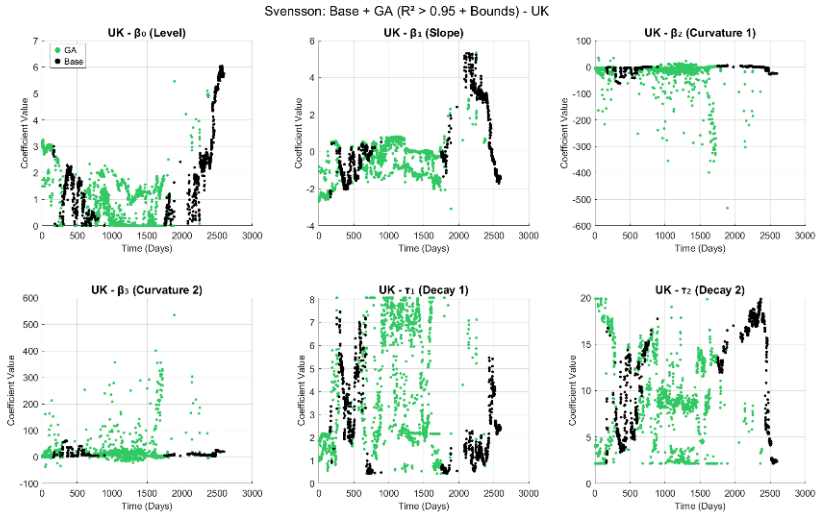


Figure A.213: Svensson Model: Base vs. GA Optimized Performance – UK

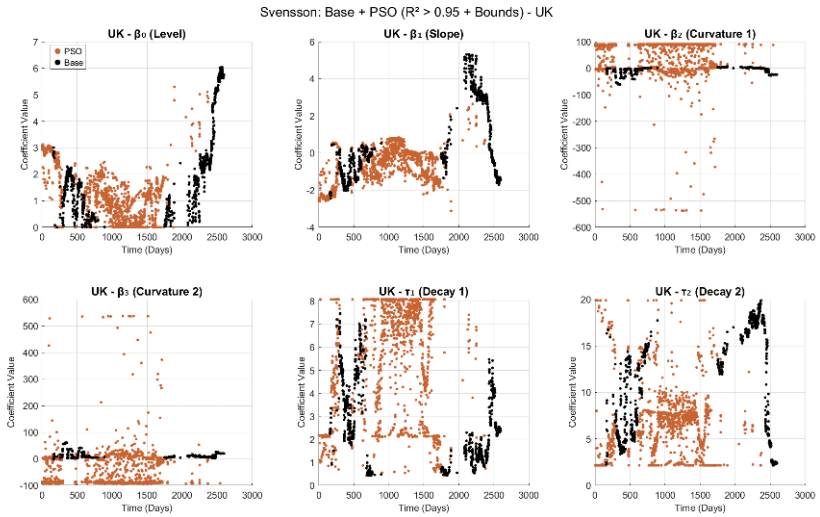


Figure A.214: Svensson Model: Base vs. PSO Optimized Performance – UK

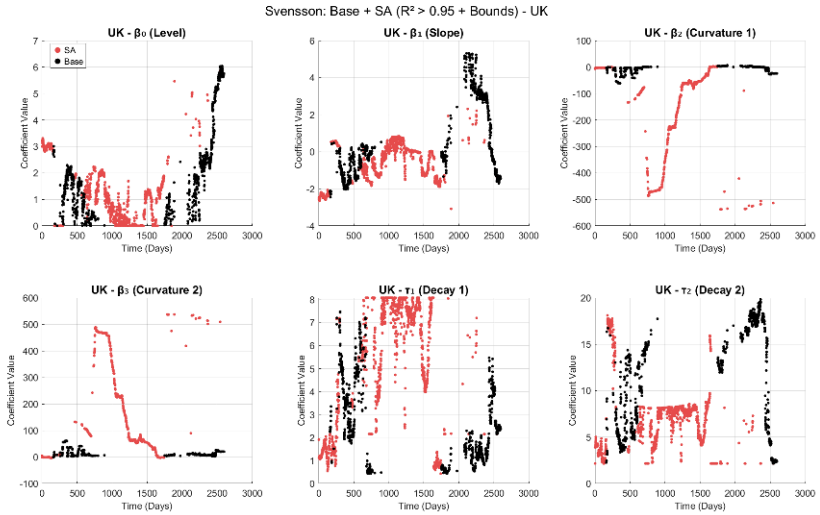


Figure A.215: Svensson Model: Base vs. SA Optimized Performance – UK

A.3.2 Cumulative Performance Comparison

The following plots show the cumulative number of yield curves resolved over time for each country and model.

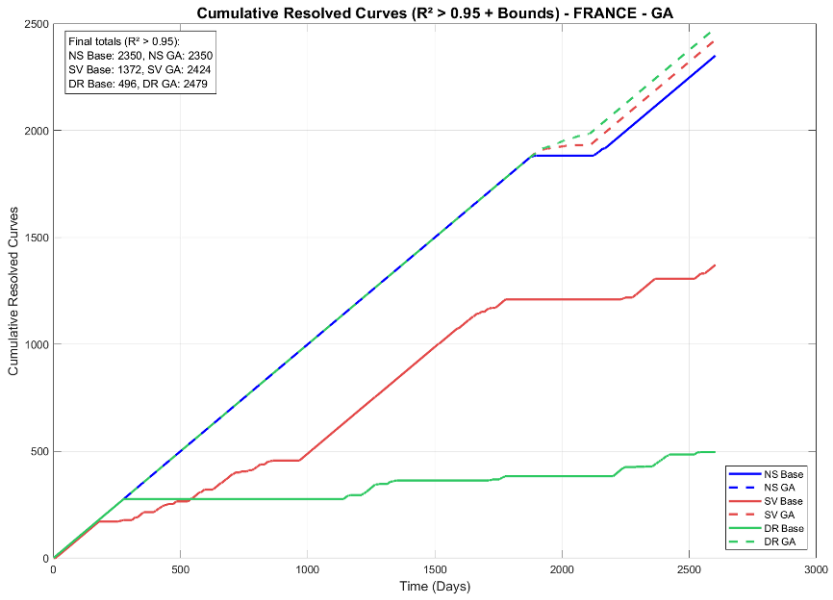


Figure A.216: Cumulative Performance (GA Optimized) – FRANCE

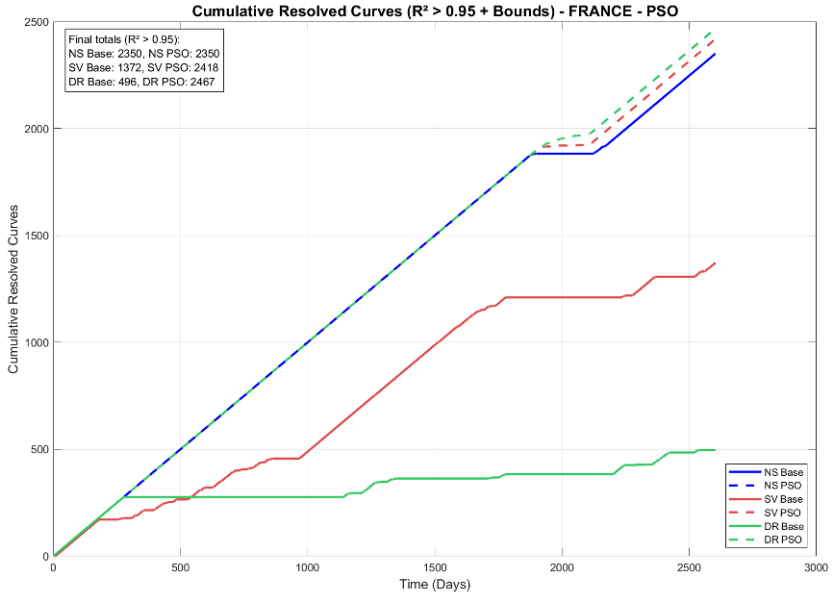


Figure A.217: Cumulative Performance (PSO Optimized) – FRANCE

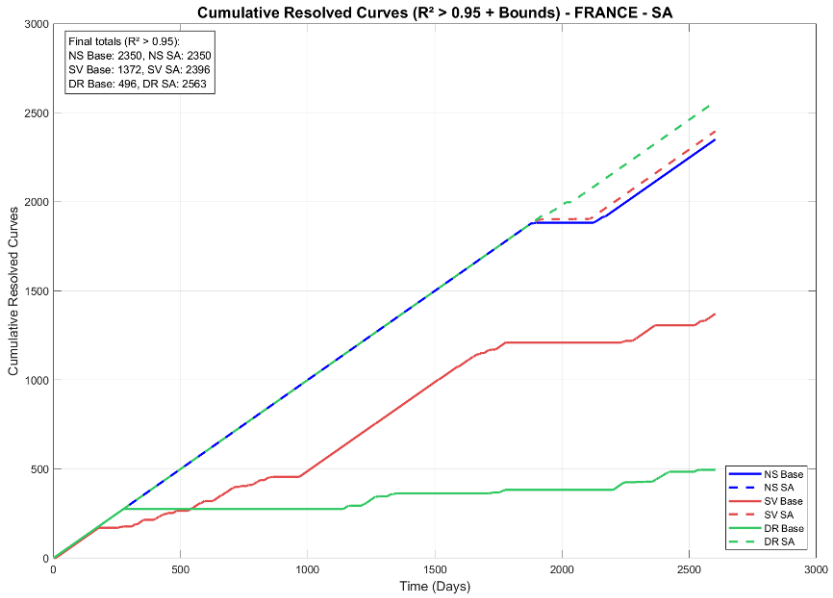


Figure A.218: Cumulative Performance (SA Optimized) – FRANCE

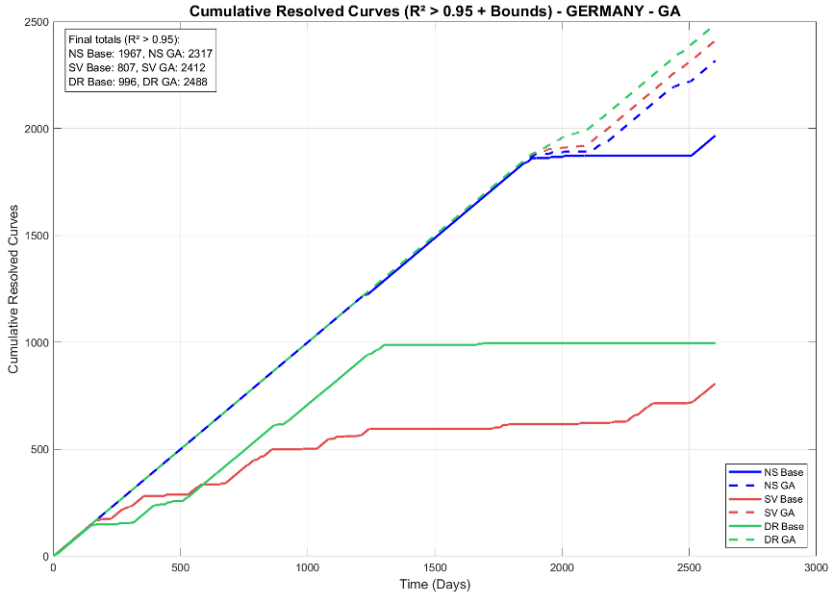


Figure A.219: Cumulative Performance (GA Optimized) – GERMANY

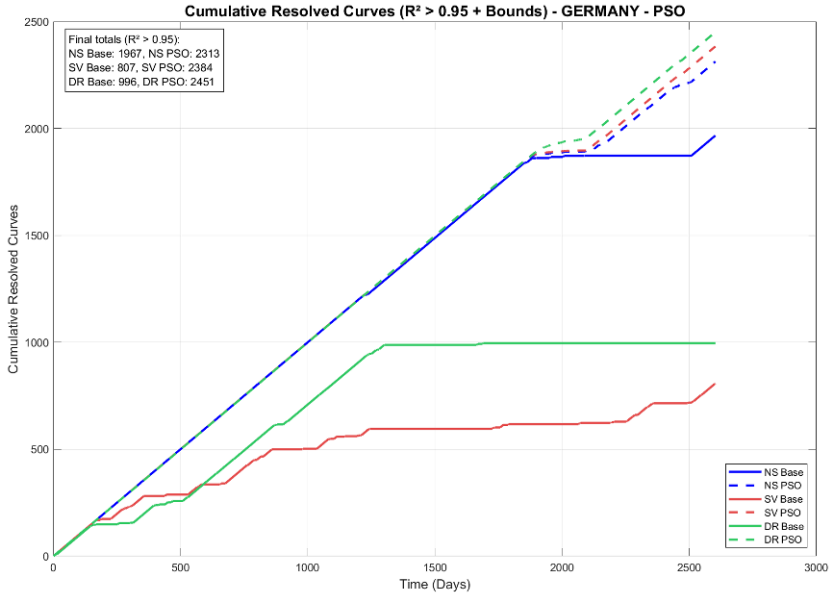


Figure A.220: Cumulative Performance (PSO Optimized) – GERMANY

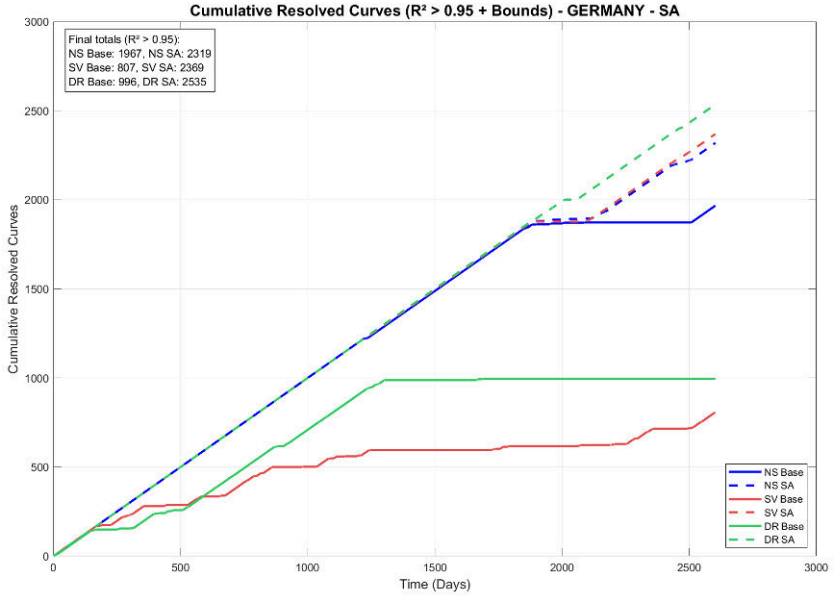


Figure A.221: Cumulative Performance (SA Optimized) – GERMANY

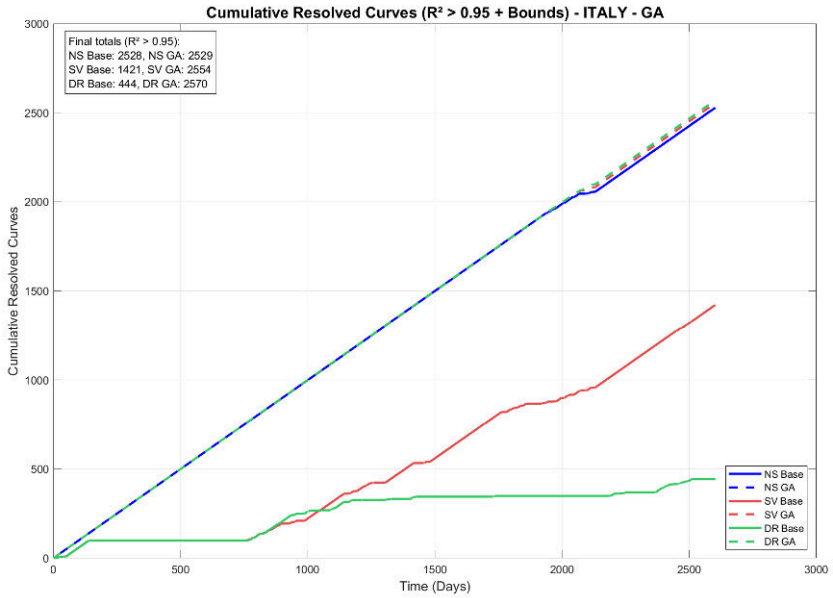


Figure A.222: Cumulative Performance (GA Optimized) – ITALY

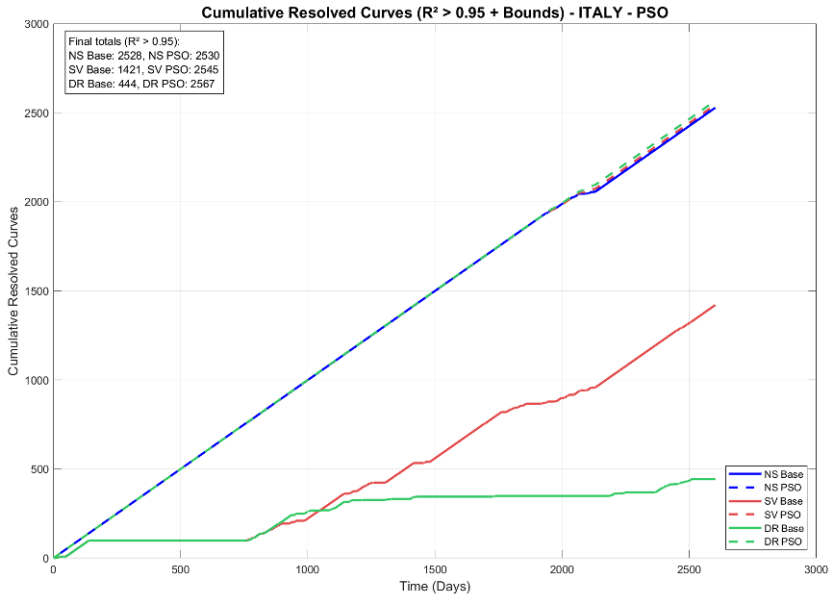


Figure A.223: Cumulative Performance (PSO Optimized) – ITALY

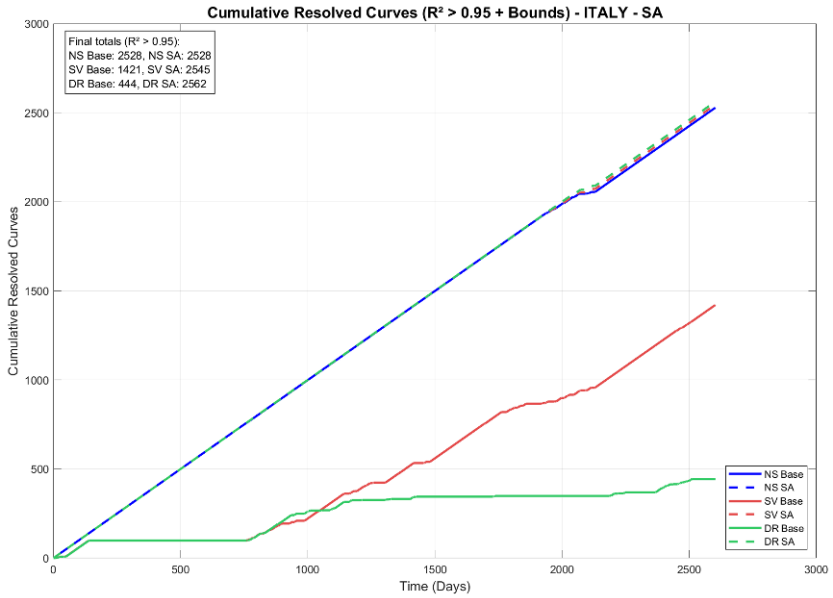


Figure A.224: Cumulative Performance (SA Optimized) – ITALY

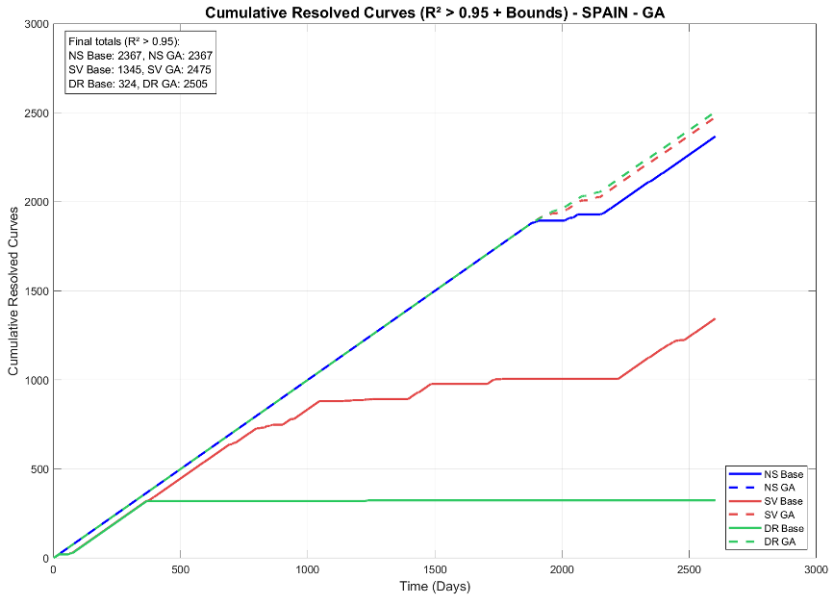


Figure A.225: Cumulative Performance (GA Optimized) – SPAIN

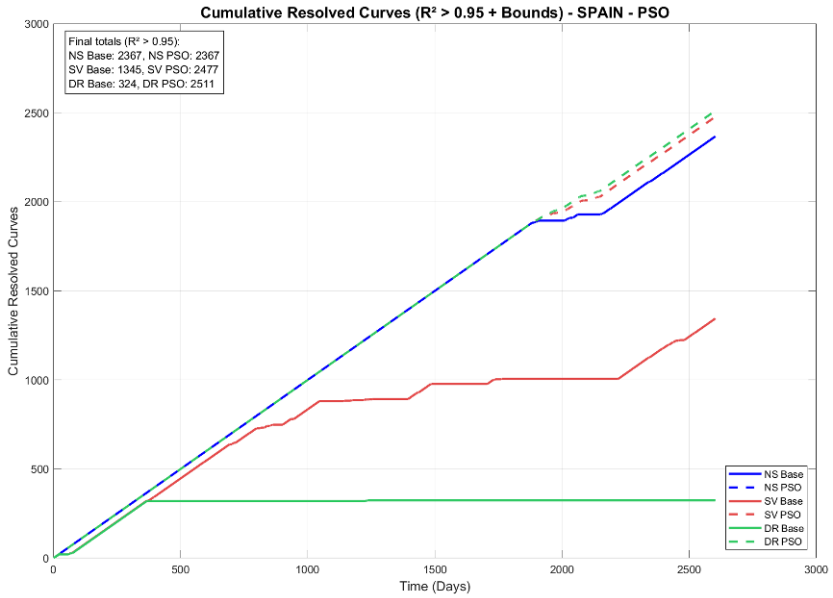


Figure A.226: Cumulative Performance (PSO Optimized) – SPAIN

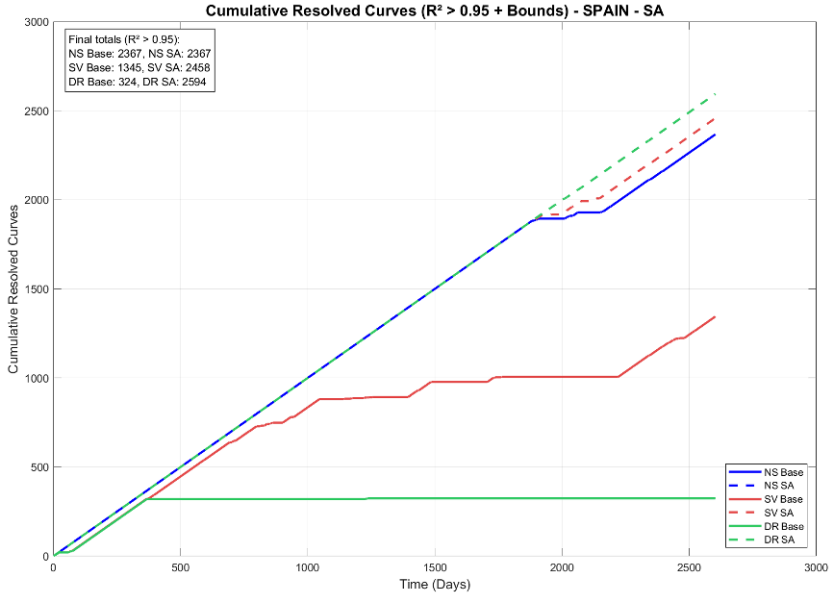


Figure A.227: Cumulative Performance (SA Optimized) – SPAIN

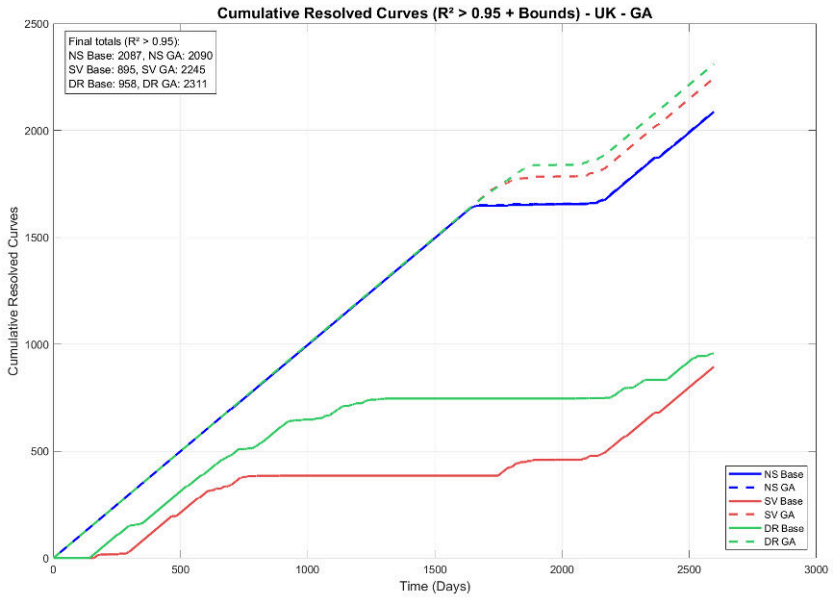


Figure A.228: Cumulative Performance (GA Optimized) – UK

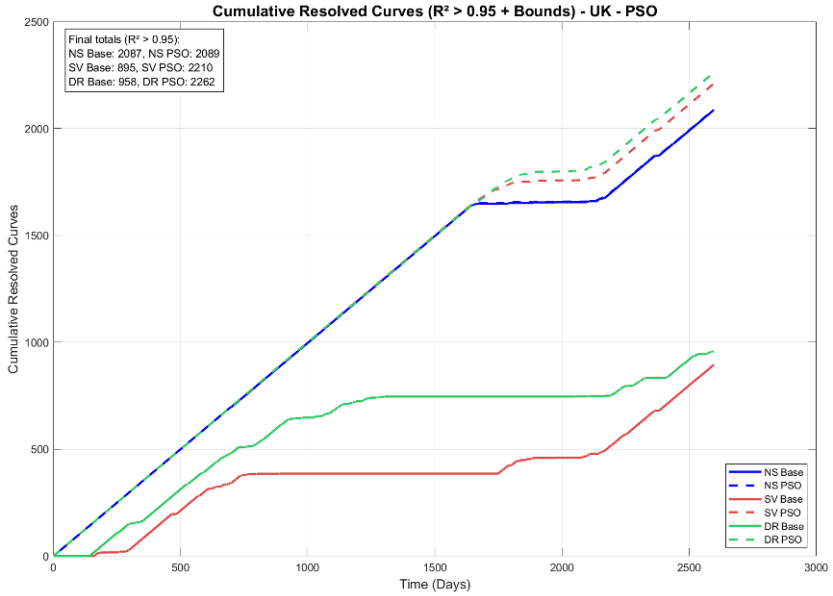


Figure A.229: Cumulative Performance (PSO Optimized) – UK

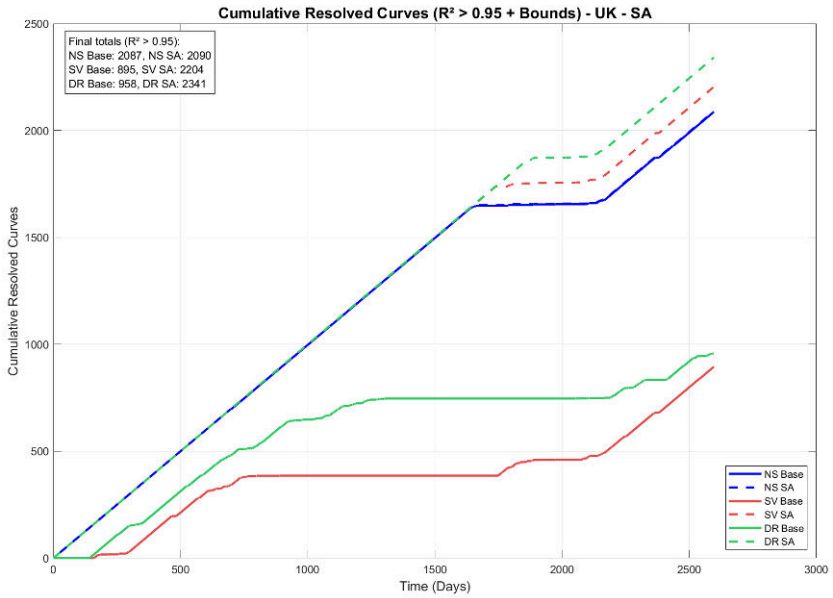


Figure A.230: Cumulative Performance (SA Optimized) – UK

A.3.3 Global Performance

The histograms that follow provide an aggregated overview of fitting success rates across all methods. The first plot shows the combined performance of all algorithms, while the subsequent ones isolate the contribution of GA, PSO, and SA respectively

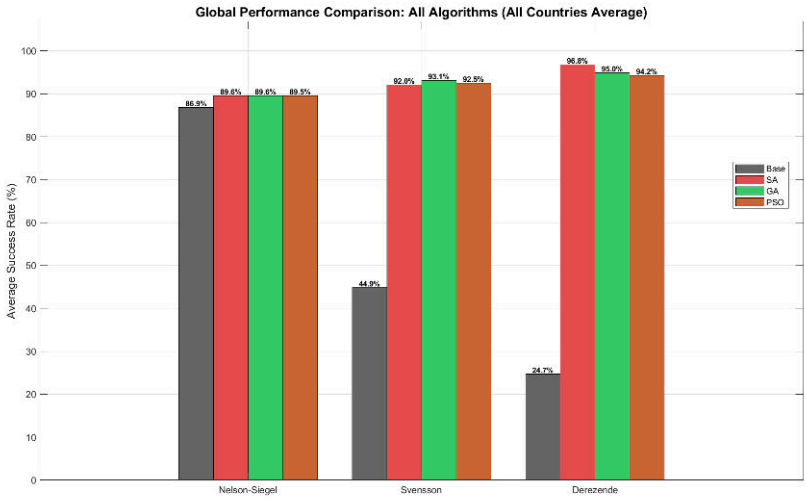


Figure A.231: Global Performance Histogram – All Algorithms

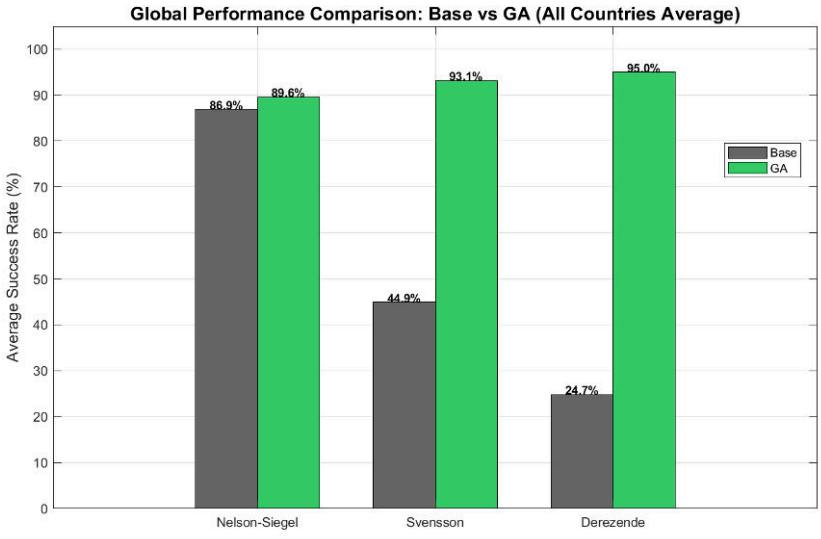


Figure A.232: Global Performance Histogram – GA

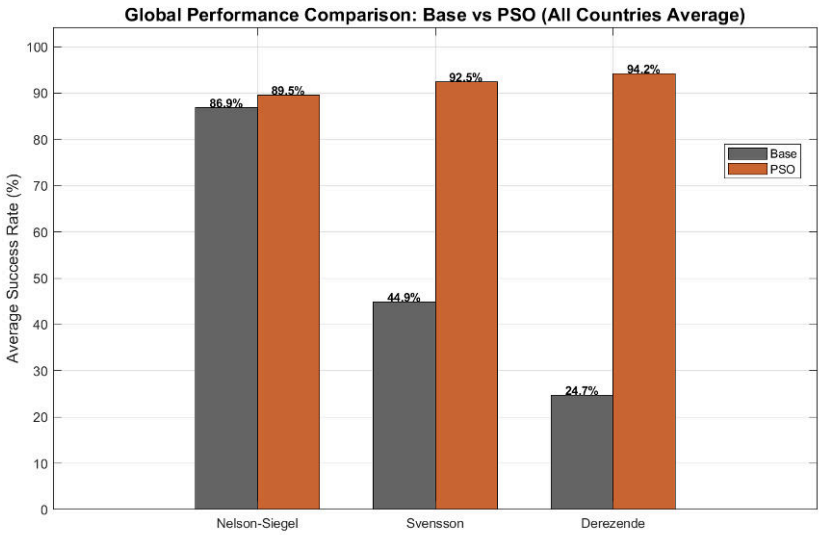


Figure A.233: Global Performance Histogram – PSO

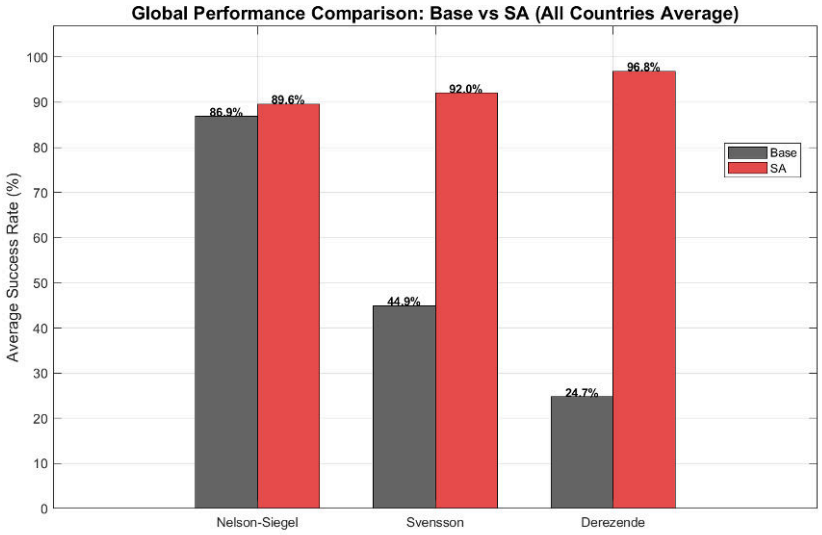


Figure A.234: Global Performance Histogram – SA

A.3.4 Vasicek Forecasts

The final set of figures depicts the short-term interest rate forecasts for each country, generated using the Vasicek model.

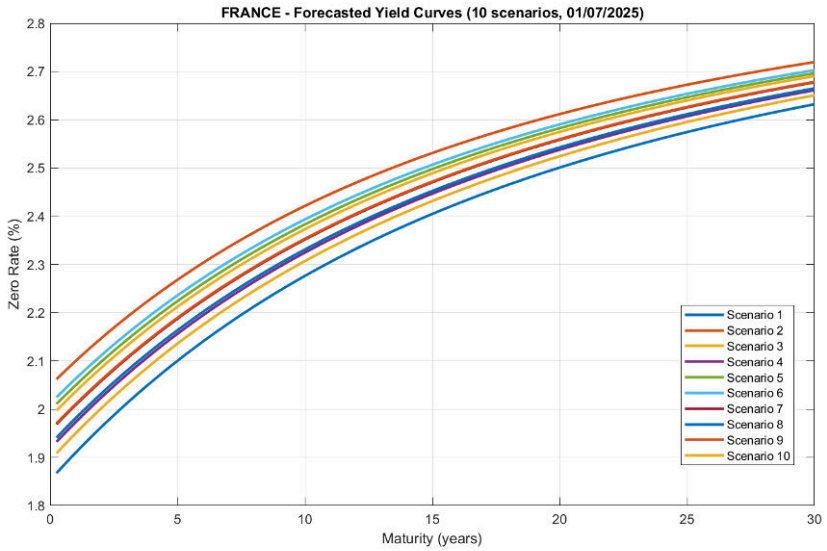


Figure A.235: Vasicek Forecast – FRANCE

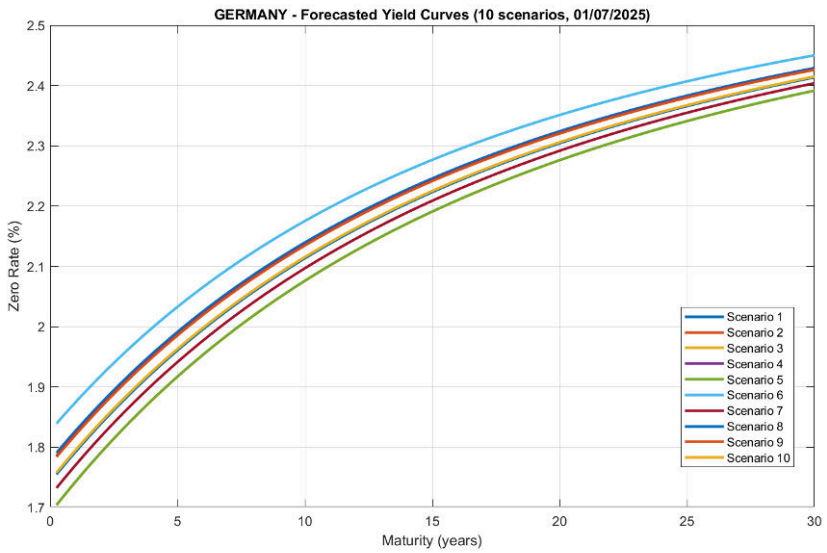


Figure A.236: Vasicek Forecast – GERMANY

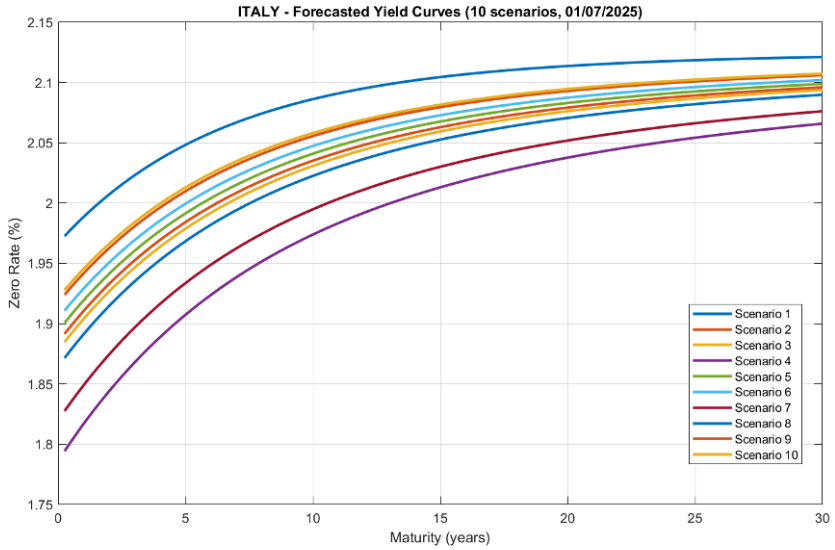


Figure A.237: Vasicek Forecast – ITALY

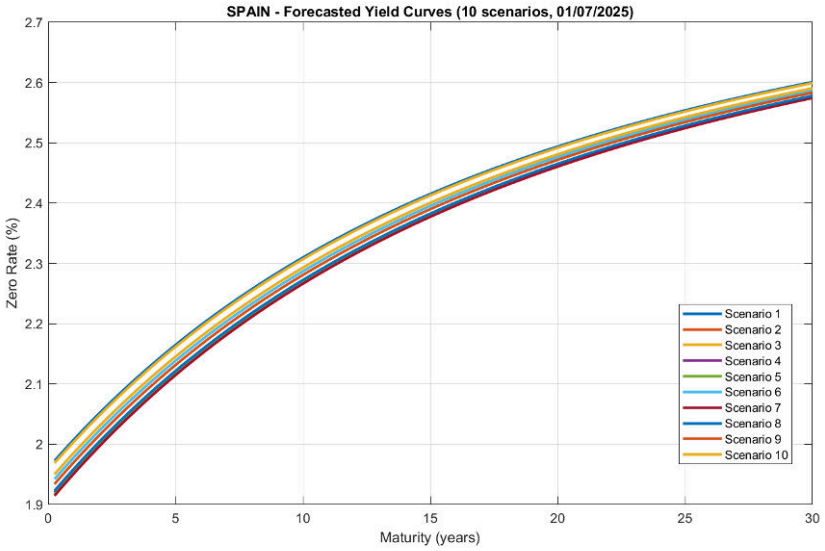


Figure A.238: Vasicek Forecast – SPAIN

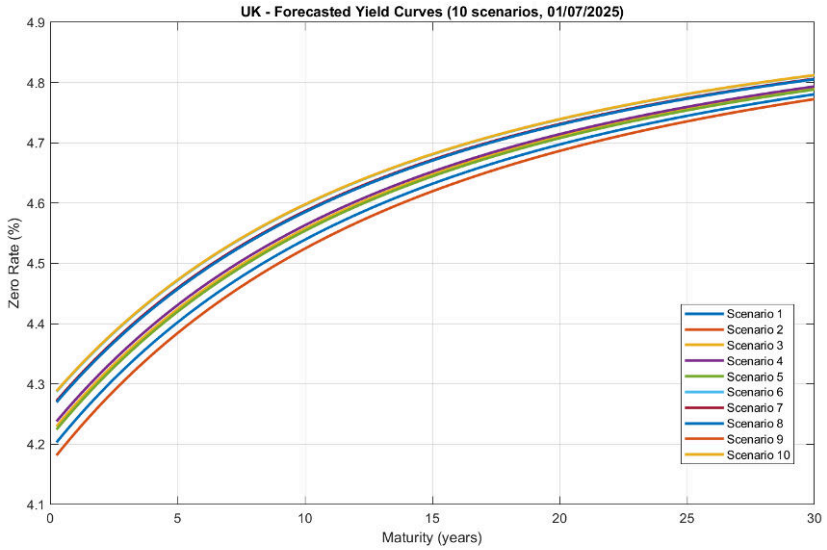


Figure A.239: Vasicek Forecast – UK

CHAPTER 3

A reliable and prudent forward-looking Value at Risk estimation applied to an Option Portfolio

Abstract

The Value at Risk (VaR) is one of the most widely used methods in the financial sector for assessing the risk associated with a portfolio over a specified period and is generally applied to portfolios composed of linear assets, where risk is more straightforward to quantify. In our study, aiming to provide an innovative contribution, we focus on nonlinear instruments such as options, which exhibit significant nonlinearity and require a different approach to risk estimation. Our objective is to calculate the most conservative Value at Risk for an option portfolio over a future time horizon, leveraging the Monte Carlo method to project the portfolio's performance and ensure a robust VaR estimation. To outline the most pessimistic scenario, we analyze key factors influencing the valuation of options, specifically volatility, drift, and dividends, using multiple estimation techniques based on historical data. By extracting only the extreme scenarios: "maximum and minimum values across different models" we ensure a comprehensive evaluation of risk exposure, performing extensive simulations to derive the worst-case VaR for the portfolio under varying market conditions.

Keywords: Value at Risk, financial risk management, nonlinear instruments, option portfolio, option pricing, Monte Carlo method, historical volatility, GARCH volatility model, ARIMA

JEL code: G17, E43, C45, C53

Acknowledgements

I gratefully acknowledge the insightful comments and constructive discussions that emerged during the presentation of this research at international conferences, including:

New Frontiers in Banking and Capital Markets 6th Edition (La Sapienza University and Luiss University, Rome, Italy, December 2–3, 2024),

and the 33rd European Conference on Operational Research (EURO 2024) (Technical University of Denmark, Copenhagen, Denmark, June 30–July 3, 2024).

I am especially thankful to the conference participants and anonymous reviewers for their constructive feedback, which has significantly improved the quality of this manuscript.

1 Introduction

The Value at Risk (VaR) is one of the most established and widely adopted risk measures in the financial industry for assessing the potential loss that a portfolio may incur over a specific time horizon. It addresses a fundamental question in risk management: What is the maximum loss that could be incurred over a certain time horizon, such that there is a very low probability, for example 1%, that the actual loss will exceed this amount? In practical terms, VaR quantifies the potential loss of a portfolio with a given confidence level, offering a concise probabilistic measure of financial risk exposure.

The importance of VaR lies in its versatility and the breadth of its applications within financial management. It plays a central role in financial risk management, as it allows for an assessment of the overall exposure of a portfolio to adverse price movements. VaR is also a fundamental tool in capital allocation, since it helps determine the amount of economic capital required to cover potential losses resulting from market fluctuations. In the regulatory context, VaR has been formally integrated into the Basel II and Basel III frameworks for calculating the minimum capital requirements of financial institutions. Finally, it is extensively used in internal reporting, providing an effective means to compare risk profiles across trading desks, asset classes, or business units.

Our objective in this study is to calculate a particularly conservative VaR through a methodological approach that focuses on the main determinants of option value "volatility, drift, and dividend" by selecting, for each factor, the most adverse possible scenarios. Previous works such as [Bagnato et al. 2021a] and [Bagnato et al. 2021b] have made significant contributions to this field by introducing metaheuristic approaches and combinations of models to produce more robust estimates of VaR and Expected Shortfall. However, these studies are based on linear portfolios composed of equities, bonds, and commodities, and therefore do not address the complexity that arises when non-linear instruments are included in the portfolio.

The present study aims to fill this gap by extending the analysis of Value at Risk to a portfolio of options, which are characterized by a strongly non-linear payoff structure. To achieve this, before performing the Monte Carlo simulation and thus calculating the VaR, we analyzed the main factors influencing option pricing: volatility, drift, and dividend, in order to construct the most pessimistic possible scenario for our portfolio. Based on data from the past five years, we computed, for each of the eleven stocks underlying the options in our portfolio, volatil-

ity using six different methodologies (Close Price, High-Low, High-Low-Close, EWMA, GARCH(1,1), and Implied Volatility), dividends using two methodologies (Historical and Implied), and drift using four methodologies (Forward Looking, Backward Looking, ARIMA, and LSTM).

Once the volatility, drift, and dividend were projected for the next sixty-five days using these various methods, we selected, for each stock and for each day, the maximum and minimum values to define the boundary conditions of our prudential analysis. On this basis, a Monte Carlo simulation of the portfolio value was performed, conducting 20,000 daily simulations over the 65-day horizon. This process was repeated four times using different combinations of volatility and dividend (max vol - max div, min vol - min div, max vol - min div, min vol - max div), resulting in a total of 80,000 price simulations for each future day across the eleven options.

The outcome of this process is a simulated projection of the portfolio's future value over a 65-day horizon. For each day, the lowest simulated value of the portfolio was selected, and on this basis the VaR was calculated. The proposed approach allows us to estimate the portfolio's future risk in an exceptionally rigorous and conservative manner, applying VaR to a projection that is already intrinsically cautious.

2 Literature Review

The modern concept of Value at Risk (VaR) emerged in the 1980s within large investment banks, as a response to the growing need for a synthetic, comparable, and standardized measure of market risk. Before its introduction, financial institutions relied on fragmented and heterogeneous systems for measuring exposure, which made consistent evaluation and effective communication of risk across different departments extremely difficult. The formalization of VaR is attributed to J.P. Morgan, which in 1994 publicly released the RiskMetrics framework, an innovative, transparent, and replicable system for quantifying market risk in complex portfolios. The main objective was to provide a single synthetic indicator of potential portfolio losses, suitable for both internal management and external reporting purposes.

The RiskMetrics methodology was based on several core assumptions: a daily updated covariance matrix of asset returns, the assumption of normally distributed returns, a one-day time horizon, and a 95% confidence level [J.P. Morgan and Reuters 1996]. By publishing the method, J.P. Morgan effectively created a de facto industry standard that profoundly influenced both operational practice, paving the way for its inclusion in regulatory frameworks [on Banking Supervision. 1996], and academic research, fostering early empirical comparisons [Hendricks 1996] as well as foundational literature reviews [Linsmeier and Pearson 2000].

The theoretical foundations of VaR are rooted in econometric advances related to volatility modeling. [Engle 1982] introduced the ARCH (Autoregressive Conditional Heteroskedasticity) model, which provided a dynamic representation of conditional variance in financial returns. This framework was later extended by [Bollerslev 1986] to GARCH models, which became crucial for producing realistic and time-consistent volatility estimates. In parallel, [Litterman 1986] enhanced risk forecasting through the use of Bayesian Vector Autoregressions (BVARs), integrating prior information into the estimation process and improving the robustness of risk predictions.

During the 1990s, authors such as [Duffie and Pan 1997] and [Lopez 1998] made significant contributions to the theoretical and practical systematization of VaR. They outlined the main computational approaches: historical simulation, variance-covariance, and Monte Carlo simulation, discussing their respective advantages and limitations. In this period, VaR became a cornerstone of market risk management and was formally incorporated into the capital adequacy frameworks defined by the Basel Committee.

Over time, VaR evolved in an attempt to overcome its empirical limitations, particularly its inability to capture tail risk and extreme market events. [Acerbi and Tasche 2002] proposed the Expected Shortfall (ES), also known as Conditional VaR (CVaR), as a more coherent and subadditive risk measure that accounts for the average loss beyond the VaR threshold. This metric is now widely recognized as a theoretically superior extension of VaR and has been incorporated into modern regulatory frameworks.

Subsequent research aimed to refine VaR estimation through more flexible and adaptive approaches. [Ang and Chen 2002] analyzed asymmetric correlations in equity portfolios, demonstrating that traditional VaR models tended to underestimate risk under such dynamics. [Casarin et al. 2013] combined Bayesian methods with traditional VaR techniques to forecast risk for VIX futures, while [Pollard 2007] and [Dieppe et al. 2018] introduced Bayesian and machine-learning-based approaches to produce more adaptive and resilient VaR estimates.

A central role in the evolution of VaR has been played by the Monte Carlo simulation, which allows for the modeling of complex interactions among correlated risk factors. This method is particularly effective for portfolios containing nonlinear instruments such as options. [Duffie and Pan 1997] were among the first to provide a detailed description of this approach, which was later enhanced by Filtered Historical Simulation (FHS) models [Gurrola-Perez and Murphy 2015] capable of capturing volatility persistence and return non-normality.

Despite its wide adoption, VaR has also been subject to substantial criticism. Authors such as [Pritsker 2006] and [Quaranta and Zaffaroni 2008] highlighted its tendency to underestimate losses under financial stress, as it ignores the magnitude of losses beyond the quantile threshold. The 2008 financial crisis further exposed these weaknesses, prompting the development of more robust measures incorporating stress testing, non-normal distributions, and regime-switching dynamics [Sarma et al. 2003; Münnix et al. 2012].

More recently, the literature has emphasized the need for adaptive and hybrid methodologies for VaR estimation. [Bottasso et al. 2019] applied a metaheuristic framework combining Monte Carlo simulations and GARCH models to achieve more robust VaR and Expected Shortfall estimates, while [Bagnato et al. 2021b] proposed a "commitment machine" capable of selecting the most prudent VaR forecast among multiple daily estimations. However, these contributions mainly focus on linear portfolios composed of equities, bonds, and commodities, and therefore do not address the specific challenges arising from nonlinear instruments.

The present study builds upon this evolving methodological landscape by extending VaR estimation to a portfolio of options, characterized by highly nonlinear payoff structures. The objective is to contribute to the ongoing debate by proposing a reliable and prudently forward-looking approach, integrating advanced estimation techniques and Monte Carlo simulations for the assessment of future risk in nonlinear portfolios.

3 Data

3.1 Stock Characteristics

Our study's portfolio is built upon eleven stocks, chosen to offer a diversified representation of the American economy across its key sectors. The analysis period spans from January 1, 2018, to March 31, 2023. This five-year time frame ensures statistical significance while being narrow enough to capture the effects of recent, high-impact market events, such as the volatility observed during the Covid-19 pandemic.

The portfolio consists of the most representative, market-capitalization-based company from each of 11 key sectors, creating an "equal weight" selection. The companies and their sectors are detailed in Table 3.1.

Sector	**Company**
Energy	Exxon Mobil (XOM)
Technology	Apple (AAPL)
Communication Services	Alphabet (GOOGL)
Health Care	Johnson & Johnson (JNJ)
Finance	JPMorgan Chase (JPM)
Consumer Discretionary	Amazon (AMZN)
Utilities	NextEra Energy (NEE)
Real Estate	American Tower Corp (AMT)
Industrial	General Electric (GE)
Non-Durable Consumer Goods	Procter & Gamble (PG)
Materials	Linde (LIN)

Table 3.1: Stock Selection by Sector.

The historical performance of these 11 stocks, normalized to a starting value of 100 on January 1, 2018, is visually represented in figure 3.1.

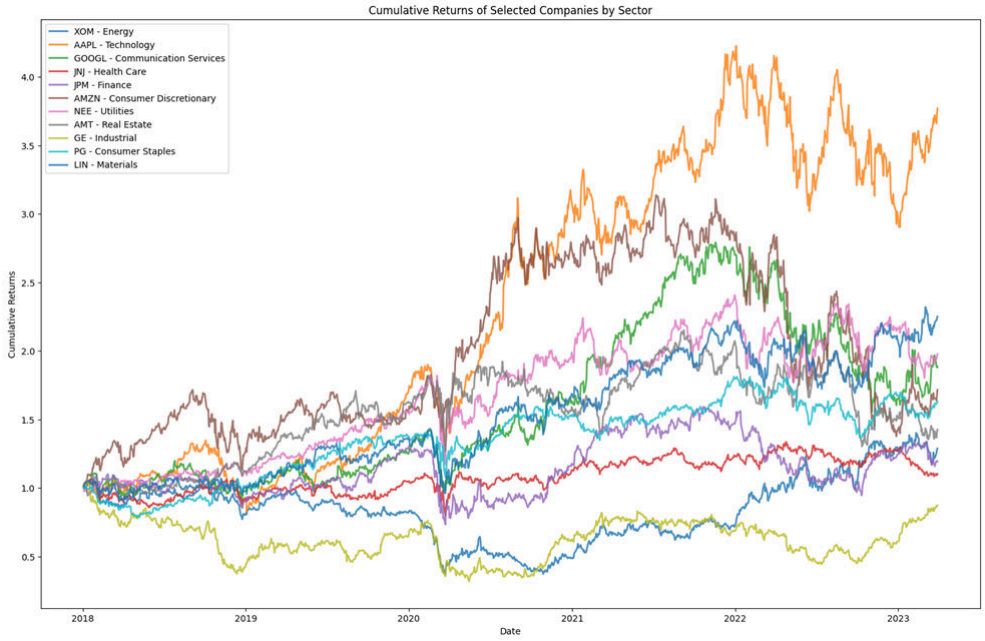


Figure 3.1: Cumulative Returns of Selected Companies by Sector.

In Table 3.2. The correlation matrix shows moderate correlations among the assets, confirming the successful implementation of diversification.

	XOM	AAPL	GOOGL	JNJ	JPM	AMZN	NEE	AMT	GE	PG	LIN
XOM	1	0.3361	0.3408	0.3139	0.5919	0.2061	0.2717	0.2717	0.5083	0.2381	0.4364
AAPL	0.3361	1	0.6867	0.4061	0.4579	0.6450	0.4216	0.4444	0.3644	0.4218	0.5383
GOOGL	0.3408	0.6867	1	0.3632	0.4523	0.6830	0.3912	0.4065	0.3364	0.3621	0.5175
JNJ	0.3139	0.4061	0.3632	1	0.4059	0.2595	0.4664	0.4806	0.2847	0.5864	0.4640
JPM	0.5919	0.4579	0.4523	0.4059	1	0.3060	0.3664	0.3888	0.5863	0.3765	0.6116
AMZN	0.2061	0.6450	0.6830	0.2595	0.3060	1	0.2994	0.3418	0.2538	0.2619	0.3733
NEE	0.2717	0.4216	0.3912	0.4664	0.3664	0.2994	1	0.6165	0.2753	0.5669	0.4718
AMT	0.2410	0.4444	0.4065	0.4806	0.3888	0.3418	0.6165	1	0.2611	0.5450	0.4737
GE	0.5083	0.3644	0.3364	0.2847	0.5863	0.2538	0.2753	0.2611	1	0.2544	0.4698
PG	0.2381	0.4218	0.3621	0.5864	0.3765	0.2619	0.5669	0.5450	0.2544	1	0.4690
LIN	0.4364	0.5383	0.5175	0.4640	0.6116	0.3733	0.4718	0.4737	0.4698	0.4690	1

Table 3.2: Correlation Matrix of Daily Stock Returns.

Table 3.3 provides the main descriptive statistics (minimum, maximum, median, mean, standard deviation, kurtosis, and skewness) for the daily return series of each stock. The high kurtosis values, particularly for JPM, NEE, AMT, and PG, indicate that the return distributions are leptokurtic (heavy-tailed) and thus deviate significantly from the normal distribution assumption, a critical finding for Value at Risk modeling.

Asset	Min	Max	Median	Mean	StdDev	Kurtosis	Skewness
XOM	-0.1304	0.1194	0.0001	0.0002	0.0213	7.8580	-0.1701
AAPL	-0.1377	0.1132	0.0012	0.0010	0.0209	7.6185	-0.2359
GOOGL	-0.1237	0.0919	0.0010	0.0005	0.0200	6.5932	-0.2177
JNJ	-0.1058	0.0769	0.0003	0.0001	0.0131	12.3446	-0.5429
JPM	-0.1621	0.1656	0.0001	0.0001	0.0201	15.1172	-0.0645
AMZN	-0.1514	0.1269	0.0012	0.0004	0.0227	7.0187	-0.1675
NEE	-0.1441	0.1283	0.0013	0.0005	0.0171	14.0364	-0.3774
AMT	-0.1644	0.1153	0.0008	0.0003	0.0181	12.4905	-0.1609
GE	-0.1644	0.1374	-0.0007	-0.0001	0.0272	7.0598	-0.1229
PG	-0.0914	0.1134	0.0007	0.0004	0.0137	13.6344	0.0161
LIN	-0.1085	0.1106	0.0010	0.0006	0.0172	8.1624	-0.0662

Table 3.3: Analysis of Stock Daily Return Descriptive Statistics.

3.2 Option Characteristics

The portfolio is comprised of 11 options underwritten on the 11 underlying stocks. To enhance diversification and introduce complexity, which is essential for a Value at Risk analysis involving nonlinear payoffs, seven distinct types of options have been selected, with four types being duplicated. This structure ensures that each option's valuation is based on different pricing models, thereby contributing to the dispersion of risk factors. The chosen options are widely recognized and used in financial markets. Table 3.4 lists the specific option types and their corresponding underlying stocks.

Option Type	Underlying Stock
Cash-or-Nothing Option	Exxon Mobil (XOM)
Plain Vanilla Option	Apple (AAPL)
Chooser Option	Alphabet (GOOGL)
Look Back Floating Strike Option	Johnson & Johnson (JNJ)
Asian Option (Turnbull and Wakeman)	JPMorgan Chase (JPM)
Standard Barrier Option	Amazon (AMZN)
Compound Option	NextEra Energy (NEE)
Plain Vanilla Option	American Tower Corp (AMT)
Asian Option (Turnbull and Wakeman)	General Electric (GE)
Cash-or-Nothing Option	Procter & Gamble (PG)
Standard Barrier Option	Linde (LIN)

Table 3.4: Option Types and Corresponding Underlying Stocks.

3.2.1 Option formulas

The following paragraphs describe the theoretical framework and pricing formulas adopted for each option type considered in the portfolio.

- **Cash-or-Nothing:**
  The Cash-or-Nothing option is a form of binary option that pays a fixed amount K at expiration if it is in the money. For a call option, the payoff is zero if the underlying price S is less than or equal to the strike price X, and K if $S > X$. For a put option, the payout is zero if $S \geq X$ and K if $S < X$. This structure provides an "all-or-nothing" payoff, depending solely on whether the underlying price crosses the strike level at expiration.

  The pricing formulas for the Call (c) and Put (p) options are given by:

$$c = Ke^{-rT}N(d) \quad \text{and} \quad p = Ke^{-rT}N(-d) \tag{3.1}$$

  where $N(\cdot)$ is the cumulative standard normal distribution function and r is the risk-free interest rate. The key parameter d is defined as follows:

$$d = \frac{\ln\left(\frac{S}{X}\right) + \left(b - \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}} \tag{3.2}$$

  where:

  - S is the current price of the underlying asset;
  - X is the strike price;
  - K is the fixed cash payoff;
  - r is the risk-free interest rate;
  - q is the continuous dividend yield;
  - b is the cost of carry (i.e., $b = r - q$);
  - σ is the volatility of the underlying asset;
  - T is the time to maturity.

- **Plain Vanilla:**
  The Plain Vanilla option represents the standard form of a European call or put option, and its fair value is determined according to the Black-Scholes model (specifically the generalized form). This model evaluates the price of an option as a function of the underlying asset price S, the strike price X, the time to maturity T, the risk-free rate r, and the volatility of the underlying asset σ. The model assumes a lognormal distribution of asset returns and an arbitrage-free market.

  The pricing formulas for the European Call (c) and Put (p) are given by:

$$\begin{aligned} c &= Se^{(b-r)T}N(d_1) - Xe^{-rT}N(d_2) \\ p &= Xe^{-rT}N(-d_2) - Se^{(b-r)T}N(-d_1) \end{aligned} \tag{3.3}$$

where $N(\cdot)$ is the cumulative standard normal distribution function. The key parameters d_1 and d_2 are defined as follows:

$$d_1 = \frac{\ln\left(\frac{S}{X}\right) + \left(b + \frac{\sigma^2}{2}\right) T}{\sigma\sqrt{T}} \tag{3.4}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{3.5}$$

- **Chooser:**
  The Chooser option provides the holder with the right to decide, at a predetermined future time, whether the contract will function as a call or as a put. This type of derivative offers flexibility, as the decision is made after observing partial market evolution. Its valuation follows the Black-Scholes framework, incorporating both the future decision time and the final expiration date. The pricing function is given by:

$$\begin{aligned} w = Se^{(b-r)T_2}N(d) - Xe^{-rT_2}N(d - \sigma\sqrt{T_2}) \\ +Se^{(b-r)T_2}N(-y) + Xe^{-rT_2}N(-y + \sigma\sqrt{T_1}) \end{aligned} \tag{3.6}$$

  where

$$d = \frac{\ln\left(\frac{S}{X}\right) + \left(b + \frac{\sigma^2}{2}\right) T_2}{\sigma\sqrt{T_2}}, \tag{3.7}$$

$$y = \frac{\ln\left(\frac{S}{X}\right) + bT_2 + \frac{\sigma^2 T_1}{2}}{\sigma\sqrt{T_1}} \tag{3.8}$$

  The model accounts for both time horizons: T_1 represents the time until the holder must decide between call and put, while T_2 denotes the total time to expiration. The parameters b and r respectively represent the cost of carry and the continuously compounded risk-free rate.

- **Lookback:**
  The Lookback Floating Strike Option provides a payoff based on the most favorable price movement of the underlying asset during the option's life. In a lookback call option, the holder can buy at the minimum observed price ($S_{\min}$), while in a lookback put option, the holder can sell at the maximum observed price ($S_{\max}$). These options allow the investor to "look back" over the life of the option to determine the optimal strike price, effectively removing the need for precise market timing.

The valuation formulas for the floating strike Call (C) and Put (P) are expressed as follows:

$$C = Se^{(b-r)T}N(a_1) - S_{\min}e^{-rT}N(a_2) + \frac{S\sigma^2}{2b}e^{-rT}\left[\left(\frac{S}{S_{\min}}\right)^{-\frac{2b}{\sigma^2}}N(a_1 - \frac{2b}{\sigma}\sqrt{T}) - e^{bT}N(a_1)\right] \tag{3.9}$$

$$P = S_{\max}e^{-rT}N(-c_2) - Se^{(b-r)T}N(-c_1) + \frac{S\sigma^2}{2b}e^{-rT}\left[-\left(\frac{S}{S_{\max}}\right)^{-\frac{2b}{\sigma^2}}N(-c_1 + \frac{2b}{\sigma}\sqrt{T}) + e^{bT}N(-c_1)\right] \tag{3.10}$$

The key parameters are defined as:

$$a_1 = \frac{\ln\left(\frac{S}{S_{\min}}\right) + \left(b + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}, \quad a_2 = a_1 - \sigma\sqrt{T} \tag{3.11}$$

$$c_1 = \frac{\ln\left(\frac{S}{S_{\max}}\right) + \left(b + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}, \quad c_2 = c_1 - \sigma\sqrt{T} \tag{3.12}$$

where:

- $S_{\min}$ and $S_{\max}$ are the minimum and maximum prices of the underlying asset observed during the option's life;
- b is the cost of carry;
- $N(\cdot)$ is the cumulative standard normal distribution function.

- **Asian:**
The Asian Option (Turnbull and Wakeman model) determines the option's value using the geometric or arithmetic average of the underlying asset's price over a specified time horizon. The [Turnbull and Wakeman 1991] model provides a semi-analytical approach to approximate the fair value of Asian options, accounting for the effects of averaging, which typically reduce volatility and the overall option premium compared to standard European options. These derivatives are particularly attractive for investors aiming to hedge or speculate on the average performance of an asset rather than its terminal value.

  The pricing formulas for the Asian Call (C) and Put (P) are derived by substituting the adjusted parameters into the generalized Black-Scholes framework:

$$
\begin{aligned}
C &= Se^{(b_A - r)T} N(d_1) - Xe^{-rT} N(d_2) \\
P &= Xe^{-rT} N(-d_2) - Se^{(b_A - r)T} N(-d_1)
\end{aligned}
\tag{3.13}
$$

The relevant relationships for the adjusted parameters are defined as:

$$
d_1 = \frac{\ln\left(\frac{S}{X}\right) + \left(b_A + \frac{\sigma_A^2}{2}\right) T}{\sigma_A \sqrt{T}}, \quad d_2 = d_1 - \sigma_A \sqrt{T} \tag{3.14}
$$

where the adjusted volatility σ_A and adjusted cost of carry b_A are:

$$
\sigma_A = \sqrt{\frac{\ln(M_2) - 2b_A T}{T}}, \quad b_A = \frac{\ln(M_1)}{T} \tag{3.15}
$$

The first and second moments (M_1, M_2) of the average distribution are calculated as:

$$
M_1 = \frac{e^{bT} - e^{bt_1}}{b(T - t_1)} \tag{3.16}
$$

$$
M_2 = \frac{2e^{(2b+\sigma^2)T}}{(b+\sigma^2)(T-t_1)^2} \left[\frac{1}{2b+\sigma^2} + \frac{e^{-(b+\sigma^2)(T-t_1)}}{b+\sigma^2} \right] \tag{3.17}
$$

where σ_A represents the adjusted volatility under the averaging process, b_A denotes the effective cost of carry, and t_1 represents the start time of the averaging period.

- **Standard Barrier:**
  The Standard Barrier Option is an exotic derivative whose payoff depends on whether the price of the underlying asset reaches a predetermined level, known as the barrier (H), during the option's life. These options are classified as knock-in (activated if H is breached) or knock-out (deactivated if H is breached).

  Since the valuation depends on the relationship between the strike price (X) and the barrier (H), the final option price is determined by a linear combination of the terms A through F. The pricing formulas for the most common Barrier Call (c) and Put (p) options (Reiner and Rubinstein, 1991) are summarized in the following table:

Option Type	Condition	Pricing Formula
Down-and-In Call (c_{di})	$X > H$	$C + E$
	$X < H$	$A - B + D + E$
Down-and-Out Call (c_{do})	$X > H$	$A - C + F$
	$X < H$	$B - D + F$
Up-and-In Call (c_{ui})	$X > H$	$A + E$
	$X < H$	$B - C + D + E$
Up-and-Out Call (c_{uo})	$X > H$	F
	$X < H$	$A - B + C - D + F$

Table 3.5: Pricing formulas for Standard Barrier Options depending on Strike (X) and Barrier (H). Puts follow symmetric rules by changing ϕ and η.

The intermediate quantities (A through F) used in the table are defined as follows:

$$A = \phi S e^{(b-r)T} N(\phi x_1) - \phi X e^{-rT} N(\phi x_1 - \phi\sigma\sqrt{T}) \tag{3.18}$$

$$B = \phi S e^{(b-r)T} N(\phi x_2) - \phi X e^{-rT} N(\phi x_2 - \phi\sigma\sqrt{T}) \tag{3.19}$$

$$C = \phi S e^{(b-r)T} \left(\frac{H}{S}\right)^{2(1+\mu)} N(\eta y_1) - \phi X e^{-rT} \left(\frac{H}{S}\right)^{2\mu} N(\eta y_1 - \eta\sigma\sqrt{T}) \tag{3.20}$$

$$D = \phi S e^{(b-r)T} \left(\frac{H}{S}\right)^{2(1+\mu)} N(\eta y_2) - \phi X e^{-rT} \left(\frac{H}{S}\right)^{2\mu} N(\eta y_2 - \eta\sigma\sqrt{T}) \tag{3.21}$$

$$E = K e^{-rT} \left[N(\eta x_2 - \eta\sigma\sqrt{T}) - \left(\frac{H}{S}\right)^{2\mu} N(\eta y_2 - \eta\sigma\sqrt{T}) \right] \tag{3.22}$$

$$F = K \left[\left(\frac{H}{S}\right)^{1+\mu} N(\eta z) + \left(\frac{H}{S}\right)^{1-\mu} N(\eta z - 2\eta\lambda\sigma\sqrt{T}) \right] \tag{3.23}$$

The auxiliary variables are defined as:

$$x_1 = \frac{\ln\left(\frac{S}{X}\right)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}, \quad x_2 = \frac{\ln\left(\frac{S}{H}\right)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T} \tag{3.24}$$

$$y_1 = \frac{\ln\left(\frac{H^2}{SX}\right)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T}, \quad y_2 = \frac{\ln\left(\frac{H}{S}\right)}{\sigma\sqrt{T}} + (1+\mu)\sigma\sqrt{T} \tag{3.25}$$

$$z = \frac{\ln\left(\frac{H}{S}\right)}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}, \quad \mu = \frac{b - \frac{\sigma^2}{2}}{\sigma^2}, \quad \lambda = \sqrt{\mu^2 + \frac{2r}{\sigma^2}} \tag{3.26}$$

where $\phi = 1$ for a Call and -1 for a Put, and $\eta = 1$ for a Down option and -1 for an Up option. K represents the rebate paid if the barrier is triggered (for knock-out) or not triggered (for knock-in).

- **Compound Options:**
  The Compound Option is a derivative of the second generation that grants the holder the right, but not the obligation, to buy (Call) or sell (Put) another option (the "underlying option") at a predetermined exercise price and time. Essentially, it is an option on an option. This structure allows investors to take leveraged positions or to hedge risks associated with volatility or future contingent events.

  There are four main configurations, depending on the nature of the compound option and the underlying option:

  - **Call on Call (cc):** Right to buy a call option.
  - **Put on Call (pc):** Right to sell a call option.
  - **Call on Put (cp):** Right to buy a put option.
  - **Put on Put (pp):** Right to sell a put option.

  The valuation is based on the Geske (1979) model, which extends the Black-Scholes framework using the bivariate cumulative normal distribution $M(\cdot,\cdot;\rho)$. The pricing formulas for the four types are:

$$\begin{aligned}
V_{cc} &= Se^{(b-r)T}M(a_1, b_1; \rho) - Ie^{-rT}M(a_2, b_2; \rho) - Xe^{-rt_1}N(a_2) \\
V_{pc} &= Xe^{-rt_1}N(-a_2) - Se^{(b-r)T}M(-a_1, b_1; -\rho) + Ie^{-rT}M(-a_2, b_2; -\rho) \\
V_{cp} &= Ie^{-rT}M(-a_2, -b_2; \rho) - Se^{(b-r)T}M(-a_1, -b_1; \rho) - Xe^{-rt_1}N(-a_2) \\
V_{pp} &= Xe^{-rt_1}N(a_2) + Se^{(b-r)T}M(a_1, -b_1; -\rho) - Ie^{-rT}M(a_2, -b_2; -\rho)
\end{aligned} \tag{3.27}$$

where the integration limits and parameters are defined as:

$$a_1 = \frac{\ln\left(\frac{S}{S^*}\right) + \left(b + \frac{\sigma^2}{2}\right) t_1}{\sigma\sqrt{t_1}}, \quad a_2 = a_1 - \sigma\sqrt{t_1} \tag{3.28}$$

$$b_1 = \frac{\ln\left(\frac{S}{I}\right) + \left(b + \frac{\sigma^2}{2}\right) T}{\sigma\sqrt{T}}, \quad b_2 = b_1 - \sigma\sqrt{T} \tag{3.29}$$

$$\rho = \sqrt{\frac{t_1}{T}} \tag{3.30}$$

In these expressions:

- S is the current asset price;
- X is the strike price of the *compound* option (paid at t_1);
- I is the strike price of the *underlying* option (paid at T);
- t_1 is the maturity of the compound option;
- T is the maturity of the underlying option ($T > t_1$);
- S^* is the critical asset price at time t_1 such that the underlying option price equals X;
- $M(\cdot, \cdot; \rho)$ is the bivariate cumulative normal distribution function.

3.2.2 Deal Caratteristics

Following the theoretical presentation of the selected option formulas, we now provide the specific contractual details for each of the 11 derivative instruments included in the portfolio.

Contract	Underlying	S	S_{min}	K	X (Payout)	T	t/t_2	t_1	H	Barrier Type	X_2	Initial Value
Plain Vanilla Call	XOM	109.66	-	115.14	-	0.75	-	-	-	-	-	8.70
Compound Put on Call	AAPL	164.90	-	171.50	-	0.50	0.50	0.25	-	-	16.49	8.17
Arithmetic Asian Call	GOOGL	103.73	-	114.10	-	0.25	-	-	-	-	-	0.66
Chooser	JNJ	155.00	-	155.00	-	0.50	0.50	0.25	-	-	-	12.88
Arithmetic Asian Put	GE	95.60	-	94.17	-	0.75	-	-	-	-	-	5.15
Lookback Call	JPM	130.31	108.59	-	-	0.75	-	-	-	-	-	60.93
Cash or Nothing Call	AMZN	103.29	-	123.95	10.329	0.75	-	-	-	-	-	2.60
Standard Barrier Call	NEE	77.08	-	2.31	-	0.75	-	-	80.93	Up & In	-	7.10
Digital Put	PG	148.69	-	118.95	14.870	0.75	-	-	-	-	-	3.89
Barrier Call	LIN	355.44	-	355.40	10.660	0.75	-	-	337.67	Down & Out	-	13.24
Plain Vanilla Put	AMT	204.34	-	194.12	-	0.75	-	-	-	-	-	13.63

Table 3.6: Portfolio Details: Maximum Disaggregation

The variables used in Table 3.6 are defined as follows:

- **S**: Current price of the underlying asset.
- **S_{min}**: Minimum price observed for the underlying asset (relevant for Lookback options).

- **K**: Strike price of the option (exercise price).
- **X** (Payout): Fixed cash payment amount (for Cash-or-Nothing/Digital options).
- **T**: Total time to expiration (in years).
- $\mathbf{t/t_2}$: Time until the final expiration date (used for Chooser and Compound options).
- $\mathbf{t_1}$: Intermediate decision time (relevant for Chooser and Compound options).
- **H**: Barrier level (relevant for Barrier options).
- **Barrier Type**: Type of barrier event (e.g., Up & In, Down & Out).
- $\mathbf{X_2}$: Strike price of the underlying option (for Compound options).
- **Initial Value**: Theoretical initial price of the derivative contract.

4 Methodology

4.1 Monte Carlo Simulation Framework and Contribution

Based on the data presented in the previous chapter, our primary objective is to create a reliable and prudent forward-looking Value at Risk (VaR) estimation. Given the highly nonlinear payoff structures of the options portfolio, we adopt the Monte Carlo (MC) simulation method. This approach allows us to generate numerous projections of the portfolio's potential future states, providing a robust distribution of possible losses.

The core of the Monte Carlo simulation for financial assets relies on modeling the price path of each underlying stock (S_t) using a **Stochastic Differential Equation (SDE)**. We model the dynamics of the underlying assets using the Geometric Brownian Motion (GBM) framework:

$$dS_t = \mu S_t dt + \sigma S_t dW_t \tag{4.1}$$

Where:

- S_t is the Stock price at time t.
- dW_t is the increment of the Wiener process (standard Brownian motion).
- μ is the instantaneous Drift (expected return).
- σ is the instantaneous Volatility.
- dt is the infinitesimal change in time.

The complexity of accurately forecasting portfolio risk lies in the correct estimation of the parameters μ and σ. Our primary contribution to the literature is the proposal of a methodology designed to yield a more prudent VaR forecast by systematically addressing the uncertainty inherent in these parameters. We evaluate the necessary parameters using multiple, varied methodologies. We then compare these estimations and strategically select the values that lead to the most conservative risk measure, thereby improving the robustness of the VaR calculation that relies on the Monte Carlo model.

4.2 Multiple Estimation Methodologies

In this paper, we employ six different methods for calculating volatility (σ) estimations and four methods for drift (μ) estimations.

4.2.1 Volatility (σ) Estimation Models

All volatility methods, with the exception of Implied Volatility, fall under the category of Historical Volatilities.

1. **Historical Volatility (Close-to-Close):**
   Close-to-Close prices is one of the most common methods for calculating historical volatility [Haug 2007]. It is calculated by taking the square root of the mean of the squared logarithmic deviations of the members of the population sample from their mean.

   The logarithmic return r_t is defined as:

   $$r_t = \log\left(\frac{P_t}{P_{t-1}}\right) \tag{4.2}$$

   Where P_t is the closing price at time t. The volatility (σ) is then calculated by:

   $$\sigma = \sqrt{\text{Var}(r_t)} \times \sqrt{D} \tag{4.3}$$

   Where $\text{Var}(r_t)$ is the variance of the logarithmic returns and D is the number of trading days in a year (usually 252).

2. **High-Low Volatility (Parkinson, 1980):**
   The High-Low volatility was introduced by [Parkinson 1980] and is known to be more efficient than the Close Prices method due to the higher number of observations implicitly used for the same time interval. It is formulated using the logarithmic range (H_t is the high price, L_t is the low price):

   $$r_t^{\text{HL}} = \log\left(\frac{H_t}{L_t}\right) \tag{4.4}$$

   The annualized volatility is given by:

   $$\sigma = \frac{\sqrt{\sum_{i=1}^{n}(r_i^{\text{HL}})^2}}{n\sqrt{2\log(2)}} \times \sqrt{D} \tag{4.5}$$

   Where n is the number of observations.

3. **High-Low-Close Volatility (Garman and Klass, 1980):**
   [Garman and Klass 1980] established the High-Low-Close volatility method, which combines high, low, and close prices for both

weekly and daily observations to further increase efficiency. The formula for the volatility (σ) is:

$$\sigma = \sqrt{\left(\frac{1}{2n}\sum\left(\log\left(\frac{H_t}{L_t}\right)\right)^2\right) - \frac{1}{n}\left(2\log(2)-1\right)\sum r_t^2} \times \sqrt{D} \tag{4.6}$$

Where r_t are the logarithmic returns based on closing prices.

4. **Exponentially Weighted Moving Average (EWMA):**
The EWMA model (Alexander, 2001) assigns exponentially decaying weights to past observations, giving more weight to the most current observations. We use the fixed weighting factor $\lambda = 0.94$, consistent with the value adopted by J.P. Morgan's RiskMetrics framework [Haug 2007]. The annualized volatility is given by the square root of the annualized variance:

$$\sigma = \sqrt{\text{ExpVol} \times D} \tag{4.7}$$

Where the variance estimate (ExpVol) is a function of the previous variance estimate ($\text{ExpVol}_{\text{prev}}$) and the current squared logarithmic return:

$$\text{ExpVol} = \lambda \times \text{ExpVol}_{\text{prev}} + (1-\lambda) \times \left(\log\left(\frac{P_{t+1}}{P_t}\right)\right)^2 \tag{4.8}$$

5. **GARCH(1,1) Model (Bollerslev, 1986):**
The Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model allows the conditional variance (ν_i) to depend on both the past squared returns and the past conditional variances. The GARCH(1,1) model for the conditional variance is:

$$\nu[i] = \omega + \alpha \times u[i-1]^2 + \beta \times \nu[i-1] \tag{4.9}$$

Where $u[i-1]$ is the most recent return (or shock), $\nu[i-1]$ is the most recent estimate of the variance, and ω is the weight assigned to the long-run average variance rate (V_L), defined as:

$$\omega = V_L \times (1-\alpha-\beta) \tag{4.10}$$

The weights must sum to unity, i.e., $\alpha + \beta + (1-\alpha-\beta) = 1$. For estimating the parameters (ω, α, β), we use Maximum Likelihood (ML) methods. The log-likelihood function to be maximized (subject to $\alpha + \beta < 1$ for stability) is:

$$\ln(L) = -\sum_{i=1}^{n}\left[\frac{1}{2}\ln(2\pi) + \frac{1}{2}\ln(\nu_i) + \frac{u_i^2}{2\nu_i}\right] \tag{4.11}$$

6. **Implied Volatility (Hull, 2021):**
Implied Volatility is a forward-looking measure derived from the market price of options using a pricing model. It reflects the market's expectation of future volatility. We utilize Implied Volatility data sourced from Bloomberg®.

4.2.2 Drift (μ) and Dividend (q) Estimation Models

The drift parameter (μ) in our SDE (Equation 4.1) is modeled under the risk-neutral measure as the risk-free rate (r) adjusted for the dividend yield (q): $\mu = r - q$. We explore four methods for estimating the drift component.

1. **Forward Looking Dividend Yield (q_{IMPL}):**
   This method [Bottasso et al. 2019] uses a market-implied dividend yield (q_{IMPL}), typically derived from futures or options data, providing a forward-looking estimate for the drift. We source q_{IMPL} data from Bloomberg®.

$$\mu = r - q_{\text{IMPL}} \tag{4.12}$$

2. **Backward Looking Dividend Yield (q_{HIST}):**
   This approach [Bagnato et al. 2021b] utilizes the historical dividend yield (q_{HIST}) calculated from past dividend payments to estimate the drift. We source the necessary dividend data from Bloomberg®.

$$\mu = r - q_{\text{HIST}} \tag{4.13}$$

3. **ARIMA Model (Tsay, 2010):**
   The Auto-Regressive Integrated Moving Average (ARIMA) model is a statistical tool used for modeling and forecasting time series data. ARIMA is defined by three parameters: ARIMA(p, d, q), where p is the order of the Autoregressive (AR) term, d is the degree of differencing required for stationarity, and q is the order of the Moving Average (MA) term. We use ARIMA to forecast the stock returns, which serve as the drift proxy.

4. **LSTM Model (Hochreiter and Schmidhuber, 1997):**
   The Long-Short Term Memory (LSTM) model is a type of recurrent neural network (RNN) particularly suited for sequence data like time series. LSTMs are effective at learning long-term dependencies, making them a robust method for predicting complex, non-linear patterns in asset returns that can be used as the drift component.

4.3 Correlation Analysis

To solve the SDE and generate all price paths within the portfolio, it is essential to consider the correlation that exists between the financial variables. This is achieved mathematically by operating on the Wiener term (dW_t), specifically in the generation of random numbers, through the implementation of the Cholesky Decomposition.

The aim, addressing the problem in an n-dimensional context (where $n = 11$ in our case), is to generate a vector of Gaussian random variables, $\vec{X}$, that takes into account a correlation structure, expressed by the covariance matrix, $\mathbf{\Lambda}$.

We start by defining $\vec{Z} = (Z_1, Z_2, \ldots, Z_n)^\top$ as an independent Gaussian random vector of zero mean and unit variance, such that $E[\vec{Z}] = \vec{0}$ and $\text{Var}[\vec{Z}] = I$, where I is the identity matrix.

The correlated vector $\vec{X}$ is derived from $\vec{Z}$ using a lower triangular matrix, L, of appropriate dimensions:

$$\vec{X} = L\vec{Z}$$

The covariance of $\vec{X}$ is then shown to be:

$$\text{cov}(\vec{X}, \vec{X}^\top) = E[\vec{X}\vec{X}^\top] = E[L\vec{Z}(L\vec{Z})^\top] = LE[\vec{Z}\vec{Z}^\top]L^\top = LIL^\top = LL^\top \tag{4.14}$$

Thus, we require $\text{cov}(\vec{X}, \vec{X}^\top) = \mathbf{\Lambda}$, which means that L is the Cholesky decomposition of the covariance matrix $\mathbf{\Lambda}$, i.e., $\mathbf{\Lambda} = LL^\top$.

We illustrate the procedure by considering a basket of $n = 2$ underlyings with a correlation coefficient ρ and volatilities σ_1 and σ_2. The variance-covariance matrix $\mathbf{A}$ is:

$$\mathbf{A} = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \tag{4.15}$$

We seek the lower triangular matrix $L = \begin{pmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{pmatrix}$ such that $LL^\top = \mathbf{A}$. By performing the multiplication $LL^\top$ and equating the resulting elements to the covariance matrix elements, we obtain:

$$a_{11}^2 = \sigma_1^2 \implies a_{11} = \sigma_1 \tag{4.16}$$

$$a_{11}a_{21} = \rho\sigma_1\sigma_2 \implies a_{21} = \rho\sigma_2 \tag{4.17}$$

$$a_{21}^2 + a_{22}^2 = \sigma_2^2 \implies a_{22} = \sigma_2\sqrt{1-\rho^2} \tag{4.18}$$

The Cholesky matrix in two dimensions is thus:

$$L = \begin{pmatrix} \sigma_1 & 0 \\ \rho\sigma_2 & \sqrt{1-\rho^2}\sigma_2 \end{pmatrix} \tag{4.19}$$

The correlated random vector $\vec{X}$ is generated by applying L to the independent Gaussian vector $\vec{Z} = (Z_1, Z_2)^\top$:

$$\vec{X} = L\vec{Z} = \begin{pmatrix} Z_1\sigma_1 \\ \rho Z_1\sigma_2 + Z_2\sigma_2\sqrt{1-\rho^2} \end{pmatrix} \tag{4.20}$$

Extension to Three Dimensions

The same algebraic problem can be extended to three dimensions ($n = 3$) using a three-dimensional covariance matrix $\mathbf{A}$:

$$\mathbf{A} = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & \rho_{13}\sigma_1\sigma_3 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \rho_{23}\sigma_2\sigma_3 \\ \rho_{13}\sigma_1\sigma_3 & \rho_{23}\sigma_2\sigma_3 & \sigma_3^2 \end{pmatrix} = LL^\top \tag{4.21}$$

By solving the simultaneous equations from $LL^\top = \mathbf{A}$ for the elements a_{ij}, we obtain the generalized structure for the 3×3 lower triangular matrix L, which incorporates all pairwise correlation coefficients (ρ_{ij}). This decomposition works equally well in the $n = 11$ dimensions of our portfolio, allowing us to generate the proper correlated Wiener process for the Monte Carlo simulation.

The correlation matrix for our 11 selected companies is visualized in figure 4.1. The moderate correlations observed across the board confirm that the portfolio is well-diversified.

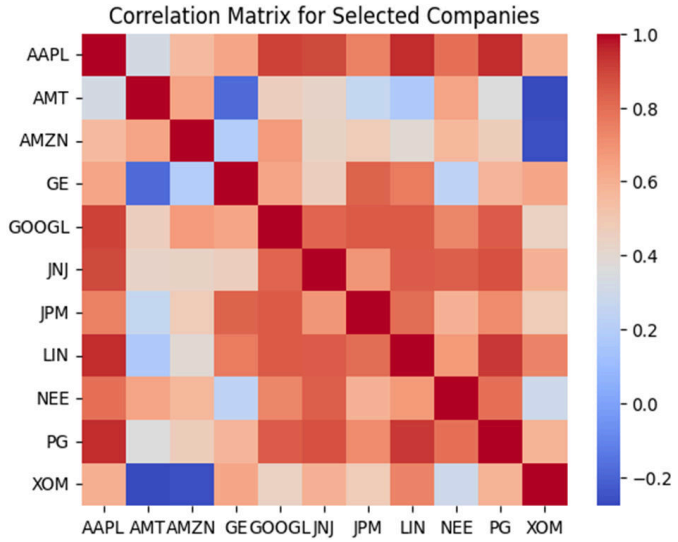


Figure 4.1: Correlation Matrix for Selected Companies

5 Results

5.1 Parameter Selection

Once the various methodologies for calculating the variables of interest were chosen, we obtained distinct values for each method. At this point, we selected the most prudent values for volatility (σ), drift (μ), and dividends (q) on a day-by-day basis.

This approach is illustrated by the figures below (figures 5.1, 5.2, 5.3, 5.4, 5.5, and 5.6), which show the maximum and minimum volatility, drift, and dividend values obtained for the Exxon Mobile (XOM) stock over time.

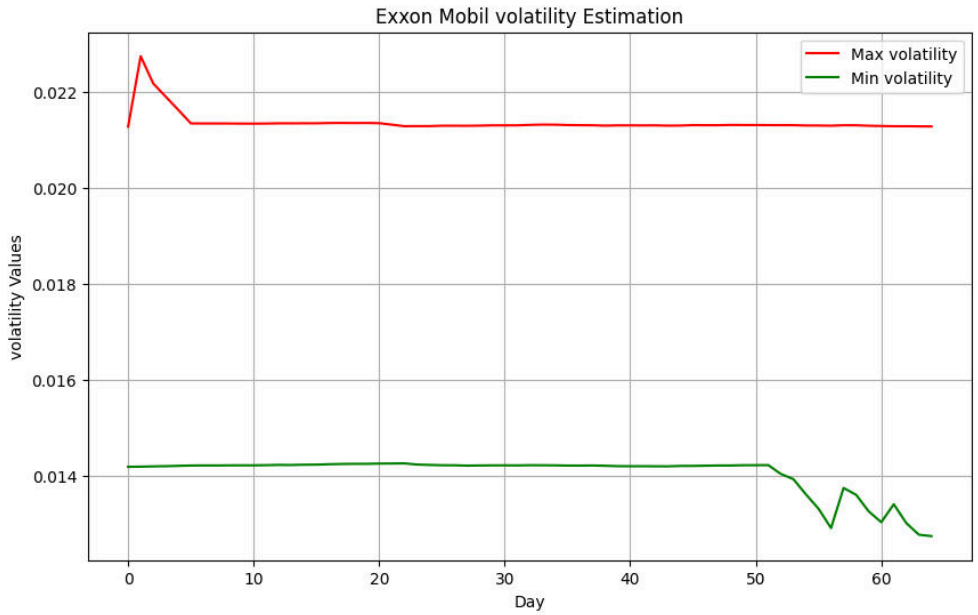


Figure 5.1: Exxon Mobil Volatility Estimation (Max/Min).

As figure 5.2 clearly demonstrates, the maximum and minimum values selected day-to-day are not necessarily obtained using a single estimation method. On the contrary, often due to changes in market direction, the optimal selection switches from one methodology to another. This is evident in the figure, where the minimum volatility curve, after a period of relative constancy (around day 50), switches between different methods as market conditions become more volatile.

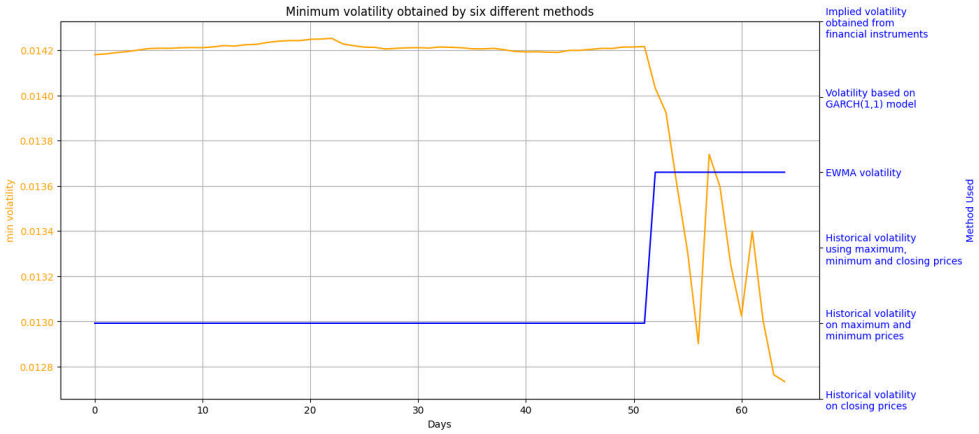


Figure 5.2: Minimum Volatility Method Used for Exxon Mobil (XOM) Over Time.

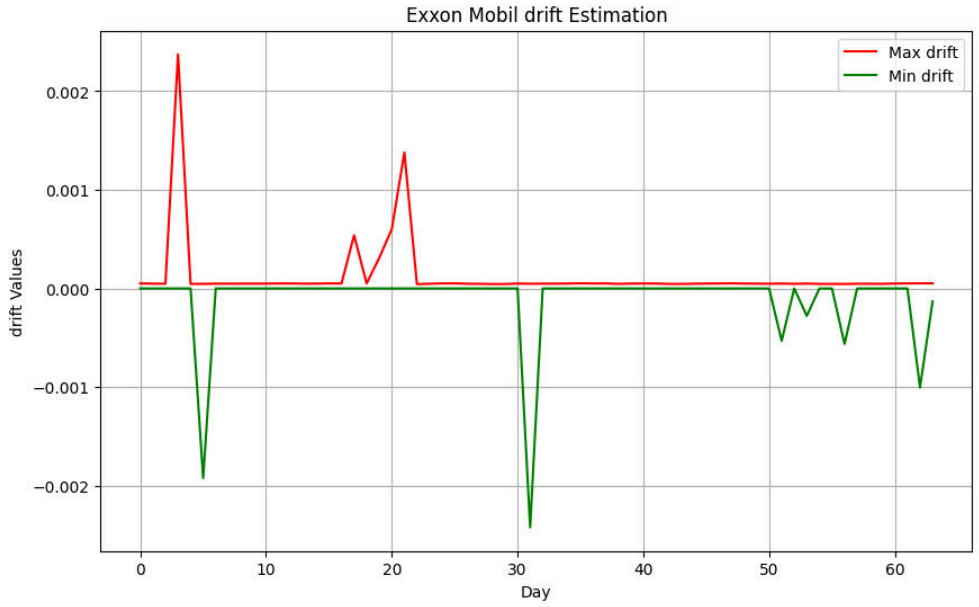


Figure 5.3: Exxon Mobil Drift Estimation (Max/Min) .

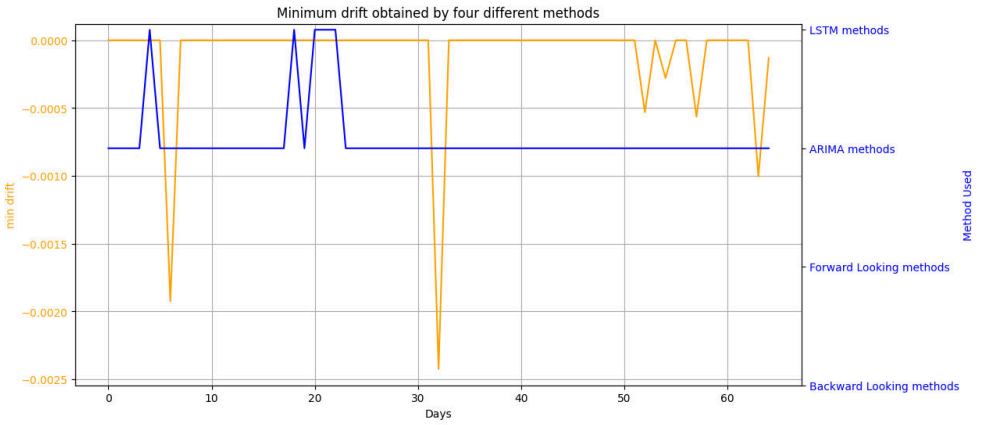


Figure 5.4: Minimum Drift Method Used for Exxon Mobil (XOM) Over Time .

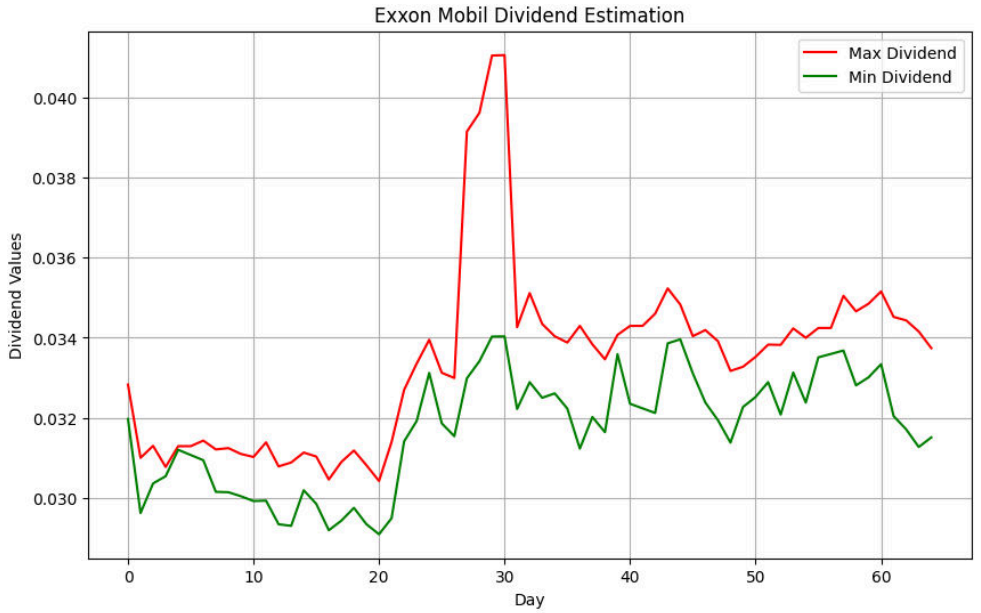


Figure 5.5: Exxon Mobil Dividend Estimation (Max/Min) .

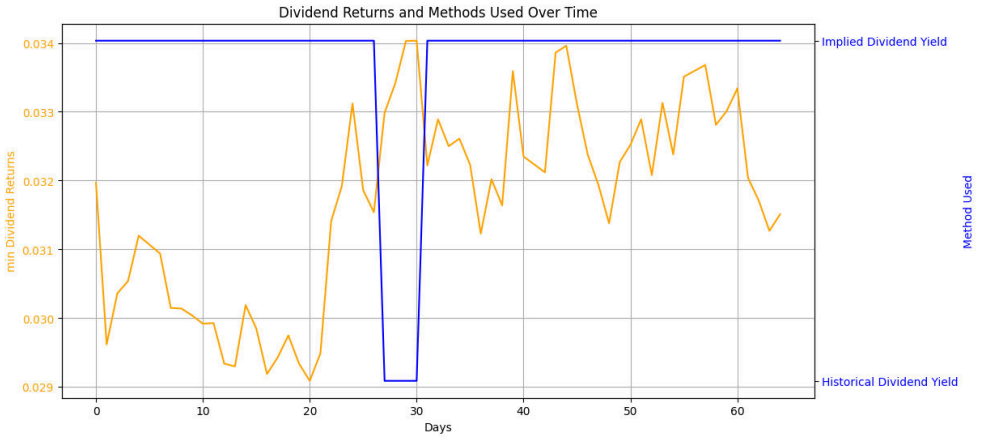


Figure 5.6: Minimum Dividend Method Used for Exxon Mobil (XOM) Over Time .

5.2 Monte Carlo Simulation

Having established the daily maximum and minimum values for our core variables of interest, which are drift (μ) and volatility (σ), we can proceed with the Monte Carlo simulation to find the VaR of our portfolio for the next 3 months (65 trading days).

The dividend yield (q) is essential for calculating two of the four drift estimation methods (Forward Looking Dividend Yield q_{IMPL} and Backward Looking Dividend Yield q_{HIST}), as $\mu = r - q$. Therefore, the dividend value is intrinsically linked to our selection of minimum/maximum drift.

To maximize the exploitation of the different extreme variable values found, we combine the maximum and minimum parameters in all possible ways. For each combination, we execute 20,000 simulations (NumSim), yielding a total of 80,000 total simulations per asset per day.

The four combined simulation scenarios are:

- 20,000 simulations: Maximum Volatility (σ_{max}), Maximum Drift (μ_{max})
- 20,000 simulations: Maximum Volatility (σ_{max}), Minimum Drift (μ_{min})
- 20,000 simulations: Minimum Volatility (σ_{min}), Maximum Drift (μ_{max})
- 20,000 simulations: Minimum Volatility (σ_{min}), Minimum Drift (μ_{min})

The simulation results for each of the 11 options yield 44 price matrices, each of size $65 \times 20,000$.

> For example, for a single option (e.g., Option 1 - Plain Vanilla), we obtain four separate matrices:
>
> $$\text{Price1}_{\text{qmin_Vmin}}, \quad \text{Price1}_{\text{qmax_Vmin}},$$
> $$\text{Price1}_{\text{qmin_Vmax}}, \quad \text{Price1}_{\text{qmax_Vmax}}$$
>
> Each matrix contains 20,000 simulated prices for each of the 65 trading days.

5.3 VaR Estimation

The final phase involves calculating the Value at Risk for each option and for the total options portfolio.

Once the price simulations are obtained from the four combined scenarios (totaling 80,000 price paths per asset per day), we calculate the VaR by finding the relevant percentiles of the final combined distribution of these simulated prices for each day. This approach ensures

that the VaR captures the risk across the entire spectrum of extreme parameter combinations ($\mu_{\max/\min}$ and $\sigma_{\max/\min}$).

We calculate the VaR at three critical confidence levels to provide a complete view of the portfolio's tail risk: the 10th percentile (90% confidence), the 5th percentile (95% confidence), and the 1st percentile (99% confidence).

1. **Individual Option VaR Calculation:** For each trading day (i in $NumLoop$), the VaR for each option (j) is calculated by finding the relevant percentile (α) of the full distribution of its 80,000 simulated prices (Price_j):

$$\text{VaR}_j[i, \alpha] = F^{-1}_{P_{j,i}}(\alpha) = \inf\{x \in \mathbb{R} : P(P_{j,i} \leq x) \geq \alpha\} \tag{5.1}$$

Note: For the first option (Var_1), the VaR calculation is based on the percentage change in price relative to the initial price, while for the others, it is based on the absolute price.*

2. **Portfolio VaR Calculation:** The total portfolio VaR (VaR_{port}) is calculated by taking the aggregate of the individual option VaR contributions for each time period:

$$\text{VaR}_{\text{port}}[i, \alpha] = \sum_{j=1}^{n} \text{VaR}_j[i, \alpha] \tag{5.2}$$

where n represents the total number of options included in the portfolio (in this specific case, $n = 11$).

This provides the final, aggregated risk measure for the entire portfolio.

The figure 5.7 illustrates the time series of the portfolio VaR at the 10th, 5th, and 1st percentiles:

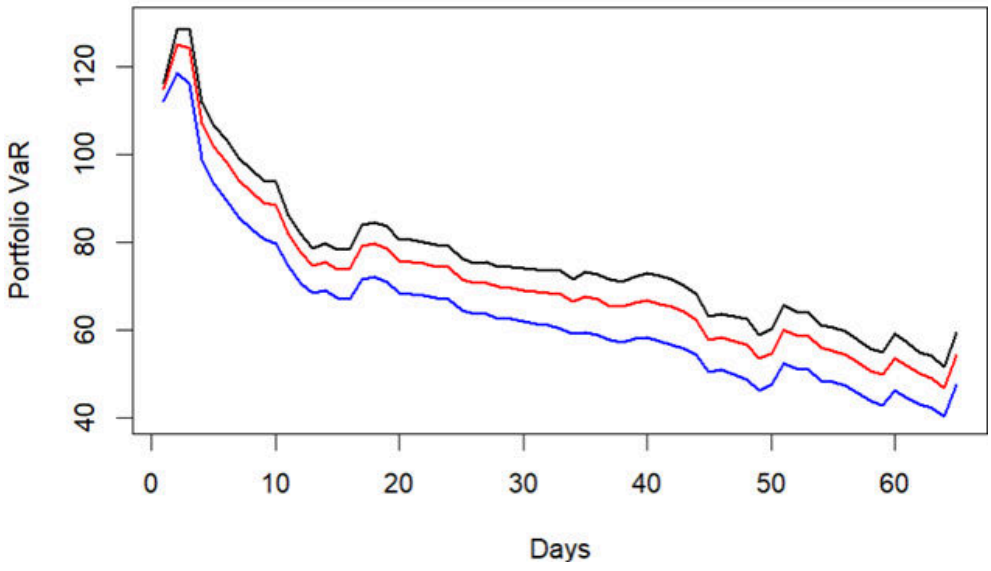


Figure 5.7: Time Series of Portfolio VaR at 10%, 5%, and 1% Percentiles .

6 Conclusions

The present study was undertaken to address the critical challenge of measuring Value at Risk (VaR) in portfolios composed of highly nonlinear instruments, such as options. The main objective was to develop an inherently robust and prudent methodology for prospective VaR estimation, integrating advanced stochastic modeling techniques.

To achieve this objective, we leveraged the Monte Carlo simulation, whose accuracy critically depends on the input parameters of volatility (σ) and drift (μ). Our methodology overcame the limitations of single-parameter models by simultaneously analyzing six methods for volatility and four for drift, selecting the most extreme (minimum and maximum) values day by day. This multi-scenario approach, which generated a total of $\mathbf{80,000}$ simulations per asset, allowed us to obtain a VaR that reflects the worst possible combination of market conditions. The final result is a set of historical series for the portfolio VaR (at the 10th, 5th, and 1st percentiles) that provides an extremely cautious and detailed view of tail risk.

The original contribution of this work lies in the introduction of a framework for the prudent and dynamic selection of parameters that feed the Stochastic Differential Equation of the Monte Carlo model. Unlike pre-existing literature, which often focuses on a single estimation model (e.g., only GARCH or only EWMA), our method explicitly combines the results of multiple heterogeneous methodologies (including Close-to-Close analysis, GARCH, EWMA, ARIMA, and LSTM) and integrates them into a multi-factorial simulation using the Cholesky Decomposition. This approach mitigates the risk of *model misspecification* and directly addresses the criticisms leveled against VaR in periods of financial stress.

The practical implications of this study are significant for risk operators and financial institutions. The methodology developed can be used as a more reliable risk management tool compared to standard models. In particular:

- **Risk Managers:** Can utilize this "prudent" VaR to allocate capital more securely for complex and nonlinear portfolios, meeting regulatory capital requirements (e.g., Basel) with an increased safety margin.
- **Fund Managers:** The framework helps to identify extreme risk scenarios that would otherwise be ignored by traditional models, enabling more targeted and effective *hedging* strategies against tail risk.

Bibliography

C. Acerbi and D. Tasche. Expected Shortfall: A Natural Coherent Alternative to Value at Risk. *Economic Notes*, 31(2):379–388, 2002. ISSN 1468-0300. doi: 10.1111/1468-0300.00091. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-0300.00091. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1468-0300.00091.

A. Ang and J. Chen. Asymmetric correlations of equity portfolios. *Journal of Financial Economics*, 63(3):443–494, March 2002. ISSN 0304-405X. doi: 10.1016/S0304-405X(02)00068-5. URL https://www.sciencedirect.com/science/article/pii/S0304405X02000685.

M. Bagnato, A. Bottasso, and P. Giribone. Design of an algorithm for an adaptive Value at Risk measurement through the implementation of robust methods in relation to asset cross-correlation. *Risk Management Magazine*, April 2021a. doi: 10.47473/2020rmm0083.

M. Bagnato, A. Bottasso, and P. G. Giribone. Implementation of a Commitment Machine for an Adaptive and Robust Expected Shortfall Estimation. *Front. Artif. Intell.*, 4, August 2021b. ISSN 2624-8212. doi: 10.3389/frai.2021.732805. URL https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2021.732805/full. Publisher: Frontiers.

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, April 1986. ISSN 0304-4076. doi: 10.1016/0304-4076(86)90063-1. URL https://www.sciencedirect.com/science/article/pii/0304407686900631.

A. Bottasso, P. G. Giribone, and M. Martorana. Analisi e progettazione di un sistema di misure quantitative per il monitoraggio dei rischi finanziari delle garanzie di origine. *Università di Genova - Dipartimento di Economia (DIEC)*, 2019.

R. Casarin, C.-L. Chang, J.-A. Jimenez-Martin, M. McAleer, and T. Pérez-Amaral. Risk management of risk under the Basel Accord: A Bayesian approach to forecasting Value-at-Risk of VIX futures. *Mathematics and Computers in Simulation*, 94:183–204, August 2013. ISSN 0378-4754. doi: 10.1016/j.matcom.2012.06.013. URL https://www.sciencedirect.com/science/article/pii/S0378475412001498.

A. Dieppe, R. Legrand, and B. van Roye. The BEAR toolbox, 2018.

D. Duffie and J. Pan. An Overview of Value at Risk. *The Journal of Derivatives*, 4(3):7–49, February 1997. ISSN 1074-1240, 2168-8524. doi: 10.3905/jod.1997.407971. URL https://www.pm-research.com/content/iijderiv/4/3/7. Company: Institutional Investor Journals Distributor: Institutional Investor Journals Institution: Institutional Investor Journals Label: Institutional Investor Journals Publisher: Portfolio Management Research Section: Primary Article.

R. F. Engle. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50(4):987–1007, 1982. ISSN 0012-9682. doi: 10.2307/1912773. URL https://www.jstor.org/stable/1912773. Publisher: [Wiley, Econometric Society].

M. B. Garman and M. J. Klass. On the Estimation of Security Price Volatilities from Historical Data. *The Journal of Business*, 53(1):67–78, 1980. ISSN 0021-9398. URL https://www.jstor.org/stable/2352358. Publisher: University of Chicago Press.

P. Gurrola-Perez and D. Murphy. Filtered Historical Simulation Value-at-Risk Models and Their Competitors, March 2015. URL https://papers.ssrn.com/abstract=2574769.

Haug. *The Complete Guide to Option Pricing Formulas*. Haug edition, 2007.

D. Hendricks. Evaluation of Value-at-Risk Models Using Historical Data. 1996.

J.P. Morgan and Reuters. RiskMetrics Technical Document – Fourth Edition, December 1996, 1996.

T. J. Linsmeier and N. D. Pearson. Value at Risk. *Financial Analysts Journal*, 56(2):47–67, March 2000. ISSN 0015-198X. doi: 10.2469/faj.v56.n2.2343. URL https://doi.org/10.2469/faj.v56.n2.2343. Publisher: Routledge _eprint: https://doi.org/10.2469/faj.v56.n2.2343.

R. B. Litterman. Forecasting With Bayesian Vector Autoregressions—Five Years of Experience. *Journal of Business & Economic Statistics*, 4(1):25–38, January 1986. ISSN 0735-0015. doi: 10.1080/07350015.1986.10509491. URL https://doi.org/10.1080/07350015.1986.10509491. Publisher: ASA Website _eprint: https://doi.org/10.1080/07350015.1986.10509491.

J. A. Lopez. Methods for Evaluating Value-at-Risk Estimates. *SSRN Journal*, 1998. ISSN 1556-5068. doi: 10.2139/ssrn.1029673. URL http://www.ssrn.com/abstract=1029673.

M. C. Münnix, T. Shimada, R. Schäfer, F. Leyvraz, T. H. Seligman, T. Guhr, and H. E. Stanley. Identifying States of a Financial Market. *Sci Rep*, 2(1):644, September 2012. ISSN 2045-2322. doi: 10.1038/srep00644. URL https://www.nature.com/articles/srep00644. Publisher: Nature Publishing Group.

Basel Committee on Banking Supervision. AMENDMENT TO THE CAPITAL ACCORD TO INCORPORATE MARKET RISKS, 1996.

M. Parkinson. The Extreme Value Method for Estimating the Variance of the Rate of Return. *The Journal of Business*, 53(1):61–65, 1980. ISSN 0021-9398. URL https://www.jstor.org/stable/2352357. Publisher: University of Chicago Press.

M. C. Pollard. Bayesian Value-at-Risk and Capital Charge Puzzle, November 2007. URL https://papers.ssrn.com/abstract=1116542.

M. Pritsker. The hidden dangers of historical simulation. *Journal of Banking & Finance*, 30(2):561–582, February 2006. ISSN 0378-4266. doi: 10.1016/j.jbankfin.2005.04.013. URL https://www.sciencedirect.com/science/article/pii/S037842660500083X.

A. G. Quaranta and A. Zaffaroni. Robust optimization of conditional value at risk and portfolio selection. *Journal of Banking & Finance*, 32(10):2046–2056, October 2008. ISSN 0378-4266. doi: 10.1016/j.jbankfin.2007.12.025. URL https://www.sciencedirect.com/science/article/pii/S0378426607004281.

M. Sarma, S. Thomas, and A. Shah. Selection of Value-at-Risk models. *Journal of Forecasting*, 22(4):337–358, 2003. ISSN 1099-131X. doi: 10.1002/for.868. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/for.868. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/for.868.

S. M. Turnbull and L. M. Wakeman. A Quick Algorithm for Pricing European Average Options. *Journal of Financial and Quantitative Analysis*, 26 (3):377–389, 1991. URL https://EconPapers.repec.org/RePEc:cup:jfinqa:v:26:y:1991:i:03:p:377-389_00. Publisher: Cambridge University Press.

A Appendix: R Code Implementation

The R code implements a quantitative framework for prudential VaR estimation of an options portfolio. The process is structured in the following main phases:

A.1 Initialization and Data Loading

Input Files:

- `23-07-06_NuovoData.xlsx` (Close, High, Low prices; implied volatilities; dividend yields; risk-free rates)

Global Parameters:

- NumLoop $= 65$ (time horizon in trading days)
- NumSim $= 20,000$ (number of Monte Carlo simulations per scenario)
- numAsset $= 11$ (number of underlying stocks)

Output: Loading of historical data for the 11 stocks over a rolling window of 1,320 days.

A.2 Volatility Estimation

Implemented Methods (6):

1. Close-to-Close: $\sigma = \sqrt{\text{Var}(\log(P_t/P_{t-1}))} \times \sqrt{252}$
2. High-Low (Parkinson, 1980)
3. High-Low-Close (Garman-Klass, 1980)
4. EWMA: $\text{ExpVol} = \lambda \times \text{ExpVol}_{\text{prev}} + (1-\lambda) \times r_t^2$, with $\lambda = 0.94$
5. GARCH(1,1): parameter estimation (ω, α, β) via Maximum Likelihood
6. Implied Volatility: data from Bloomberg

Prudential Selection:

- For each day i and asset j: $\sigma_{\max}[i,j] = \max_{m=1,\ldots,6}(\sigma_m[i,j])$ and $\sigma_{\min}[i,j] = \min_{m=1,\ldots,6}(\sigma_m[i,j])$

Output Files:

- `Volatility_log/sigma_max_out.csv`
- `Volatility_log/sigma_min_out.csv`
- `Volatility_log/max_method_per_stock.csv`
- `Volatility_log/min_method_per_stock.csv`

A.3 Dividend Yield Estimation

Implemented Methods (2):

1. Historical Dividend Yield: from Bloomberg historical data
2. Implied Dividend Yield: from Bloomberg options market data

Prudential Selection:

- $q_{\max}[i,j] = \max(q_{\text{hist}}[i,j], q_{\text{impl}}[i,j])$
- $q_{\min}[i,j] = \min(q_{\text{hist}}[i,j], q_{\text{impl}}[i,j])$

Output Files:

- `DivYield_Log/q_matrix_max.csv`
- `DivYield_Log/q_matrix_min.csv`

A.4 Drift Estimation

Implemented Methods (4):

1. Backward Looking: $\mu = r - q_{\text{hist}}$
2. Forward Looking: $\mu = r - q_{\text{impl}}$
3. ARIMA: forecasting via `auto.arima()`
4. LSTM: forecasting via recurrent neural network

Prudential Selection:

- $\mu_{\max}[i,j] = \max(\mu_{\text{backward}}, \mu_{\text{forward}}, \mu_{\text{ARIMA}}, \mu_{\text{LSTM}})$
- $\mu_{\min}[i,j] = \min(\mu_{\text{backward}}, \mu_{\text{forward}}, \mu_{\text{ARIMA}}, \mu_{\text{LSTM}})$

Output Files:

- `Drift_Log/drift_array_max.csv`
- `Drift_Log/drift_array_min.csv`

A.5 Correlation Estimation and Cholesky Decomposition

Procedure:

1. For each day i: compute correlation matrix RHO from historical returns
2. Cholesky Decomposition: RHO $= M \times M^T$, where M is lower triangular matrix
3. Verification test: $\sum(M \times M^T - \text{RHO}) \approx 0$

Output Files (for each day i):

- `Correlation_Log/RHO_matrix_i.csv`
- `Correlation_Log/M_matrix_i.csv`

A.6 Option Definition

Implemented Pricing Functions:

- `getPlainVanillaPrice()`: Plain Vanilla Call/Put options
- `getCashorNothingPrice()`: Cash-or-Nothing options
- `getChooserOptionPrice()`: Chooser options
- `getLookbackFloatingStrikePrice()`: Lookback options
- `asian_option_tw()`: Asian options (Turnbull-Wakeman)
- `standard_barrier_price()`: Barrier options (Up&In, Down&Out)
- `CompoundOptionPrice()`: Compound options (Put-on-Call)

Contract Parameters: Strike K, Spot S_0, Maturity T, Barrier H, etc. (as per Table 3.3 in the main document)

A.7 Multi-Scenario Monte Carlo Simulation

Procedure for each day i and simulation j:

1. Generate independent Gaussian shock: $\epsilon \sim N(0, I_{11})$
2. Apply correlation via Cholesky: ALPHA $= M_i \times \epsilon$

3. Compute future prices for 4 scenarios:

$$S_{\text{qmin_Vmin}}[i,j,k] = S_0[k] \cdot \exp\Big((\mu_{\min}[i,k])t + \sigma_{\min}[i,k]\text{ALPHA}[k]\sqrt{t}\Big)$$

$$S_{\text{qmax_Vmin}}[i,j,k] = S_0[k] \cdot \exp\Big((\mu_{\max}[i,k])t + \sigma_{\min}[i,k]\text{ALPHA}[k]\sqrt{t}\Big)$$

$$S_{\text{qmin_Vmax}}[i,j,k] = S_0[k] \cdot \exp\Big((\mu_{\min}[i,k])t + \sigma_{\max}[i,k]\text{ALPHA}[k]\sqrt{t}\Big)$$

$$S_{\text{qmax_Vmax}}[i,j,k] = S_0[k] \cdot \exp\Big((\mu_{\max}[i,k])t + \sigma_{\max}[i,k]\text{ALPHA}[k]\sqrt{t}\Big)$$

4. Price the 11 options in each of the 4 scenarios using the pricing functions

Intermediate Output: 44 option price matrices (11×4 scenarios), each of size $65 \times 20,000$

A.8 Prudential Aggregation

Procedure:

- For each option k, day i, and simulation j:

$$\text{Price}_k[i,j] = \min \begin{pmatrix} \text{Price}_{k_\text{qmin_Vmin}}[i,j], \\ \text{Price}_{k_\text{qmax_Vmin}}[i,j], \\ \text{Price}_{k_\text{qmin_Vmax}}[i,j], \\ \text{Price}_{k_\text{qmax_Vmax}}[i,j] \end{pmatrix}$$

Rationale: The most conservative scenario (minimum price) is selected to ensure a prudential VaR estimate.

Output Files:

- `portfolio_path/min_daily_prices.csv` (matrix 65×11 with daily minimum prices)

A.9 VaR Computation

Confidence Levels:

- 90%: VaR at 10th percentile
- 95%: VaR at 5th percentile
- 99%: VaR at 1st percentile

Procedure:

1. For each option k and day i:

$$\text{VaR}_{k,\alpha}[i] = \text{quantile}\left(\text{Price}_k[i,\cdot], \alpha\right), \quad \alpha \in \{0.01, 0.05, 0.10\}$$

2. Portfolio VaR:

$$\text{VaR}_{\text{port},\alpha}[i] = \sum_{k=1}^{11} \text{VaR}_{k,\alpha}[i]$$

Final Output: Time series of portfolio VaR for the 65 future days at the three confidence levels (as shown in figure 5.8 of the main document).

Ringraziamenti

Sono profondamente grato di aver avuto l'opportunità e il privilegio di dedicarmi allo studio approfondito di tematiche di mio grande interesse durante il conseguimento del dottorato di ricerca. Il bagaglio culturale, esperienziale e relazionale che porto con me da questa esperienza è di gran lunga superiore alle più rosee aspettative. È stato un percorso entusiasmante, vissuto costantemente alla ricerca della conoscenza e dei suoi confini inesplorati.

Questo viaggio mi è portato da Genova per poi toccare molte tappe: Offanengo, Perugia, Milano, Copenaghen, Bari, Roma, Keele, Oxford e Leeds. Non è stato un viaggio solitario: durante il tragitto ho avuto il piacere di conoscere tanti altri esploratori innamorati della cultura e della ricerca, con i quali ho condiviso un'esperienza unica e indimenticabile.

Arrivato alla fine di questo viaggio, sento il desiderio di ringraziare le diverse persone che mi hanno aiutato in modo determinante.

Innanzitutto ringrazio i miei genitori: è stata sicuramente la loro passione per la cultura, nelle sue diverse forme, a far nascere in me la scintilla che mi ha portato a interessarmi allo studio e alla ricerca della conoscenza. Desidero poi ringraziare il professor Claudio Piga, che mi ha selezionato ed è sempre stato presente per me e per i miei colleghi durante questi tre anni. Un ringraziamento speciale va al professor Pier Giuseppe Giribone, che mi ha fatto da supervisor e al quale devo tutto ciò che ho imparato durante questo dottorato. Ricordo che, quando ho iniziato questo percorso, avevo la speranza di ampliare le mie conoscenze finanziarie; ma, per quanto ottimista, non avrei mai immaginato di poter incontrare un maestro che mi guidasse in questo cammino come ha fatto il professor Giribone. Non ho mai conosciuto un docente più competente, ma soprattutto non ho mai conosciuto un professore che vivesse la ricerca non come un lavoro, ma come una passione, e che trattasse i suoi collaboratori non come colleghi, ma come amici.

Infine, ringrazio la Vhit, in quanto è anche grazie a loro che ho potuto conseguire questo dottorato. Mi auguro che in futuro siano sempre di più le aziende illuminate come la Vhit, che credono nella possibilità della collaborazione tra imprese e università e che la promuovono investendo insieme nella ricerca.