

國立臺東大學資訊管理學系
碩士論文

Department of Information Science and Management Systems

National Taitung University

Master Thesis

指導教授：廖國良 博士

Advisor : Gwo-Liang Liao, Ph. D

具反雙曲正弦函數特性之不完美除錯
軟體可靠度成長模型

**Imperfect Debugging Software
Reliability Growth Model with Inverse
Hyperbolic Sine Function**

研 究 生：湯維豪 撰

Graduate Student : Wei-Hao Tang

中華民國一零九年六月

June, 2020



國立臺東大學資訊管理學系
碩士論文

Department of Information Science and Management Systems
National Taitung University
Master Thesis

具反雙曲正弦函數特性之不完美除錯
軟體可靠度成長模型
**Imperfect Debugging Software
Reliability Growth Model with Inverse
Hyperbolic Sine Function**

研究生：湯維豪 撰
Graduate Student：Wei-Hao Tang

指導教授：廖國良 博士
Advisor：Gwo-Liang Liao, Ph. D

中華民國一零九年六月
June, 2020

國立臺東大學
學位論文考試委員審定書
系所別：資訊管理學系

本班 湯維豪 君

所提之論文 具反雙曲正弦函數特性之不完美除錯軟體可靠度成長模型

業經本委員會通過合於 碩士學位論文 條件

論文學位考試委員會：

黃建裕

(學位考試委員會主席)

謝昆彰

廖國良

(指導教授)

論文學位考試日期：109年6月22日

國立臺東大學

附註：1. 本表一式二份經學位考試委員會簽後，正本送交系所辦公室及註冊組或進修部存查。

2. 本表為日夜學制通用，請依個人學制分送教務處或進修部辦理。

國立臺東大學 學位論文授權書

重要事項說明：依著作權法第十五條第二項第三款規定，「依學位授予法撰寫之碩士、博士論文，著作人已取得學位者，推定著作人同意公開發表其著作」。本校圖書資訊館就紙本學位論文之閱覽服務依前開規定，採公開閱覽為原則。如論文涉及專利申請、投稿論文、機密或其他法定事由，需延後公開紙本論文者，請另行填寫本校「學位論文延後公開申請書」。(申請書得自本館網站下載)

本授權書所授權之論文為本人在 國立臺東大學 資訊管理 系(所)

組 108 學年度第 二 學期取得 (☒ 碩士 ☐ 博士) 學位之論文。

論文名稱：具反雙曲正弦函數特性之不完美除錯軟體可靠度成長模型

- 一、本人具有著作財產權之上列 (☒ 學位論文 ☐ 書面報告 ☐ 技術報告 ☐ 專業實務報告) 之電子全文(含書目、摘要、圖檔、影音資料、附件等)，依著作權法規定，非專屬、無償授權予下列單位得重製、上載網站，藉由網路傳輸，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

單位	公開上網時程
本人畢業學校	(依據 108 學年度第一學期第 3 次行政會議決議：研究生畢業論文延後公開上網時程，至多以三年為原則) ☐ 立即公開 ☐ 一年後公開 ☐ 二年後公開 ☒ 三年後公開
國家圖書館	☐ 立即公開 ☐ 一年後公開 ☐ 二年後公開 ☒ 三年後公開 ☐ 不同意公開
與本人畢業學校圖書資訊館簽訂合作協議之資料庫業者	☐ 立即公開 ☐ 一年後公開 ☐ 二年後公開 ☐ 三年後公開 ☒ 不同意公開

- 二、本人 (☐ 同意 ☒ 不同意) 本人畢業學校圖書資訊館基於學術傳播之目的，在上述範圍內得再授權第三人進行資料重製。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未勾選，本人同意視同同意授權。

學 號：10701304

(務必填寫)

研究生簽名：

溫維豪

(親筆正楷)

指導教授簽名：

廖國良

(親筆簽名)

日 期：中華民國 109 年 6 月 29 日

本授權書(得自本校圖書資訊館網站下載)請以黑筆撰寫並影印裝訂於書名頁之次頁。

授權書版本:2020/01/02

謝誌

時光飛逝，短暫的碩士生涯很快的過去了，感謝兩年間指導教授廖國良老師細心、用心的指導才能讓本篇論文順利完成，也感謝口試委員謝昆霖老師與黃建裕老師提供建議，使得論文整體更加的完善。感謝台東大學提供資源與環境讓我順利完成碩士學業。

感謝所有資管系和外系曾經幫助我的老師，感謝哲宇學長、濱壕學長以及其他學長姐提供給我論文的幫助，感謝仁豪、凱竹和哲言在我遇到研究瓶頸時，陪我吃飯、聊天抒發心情。

短暫的兩年曾經歡笑、曾經失落，感謝台東這片土地，感謝大山，感謝大海，感謝一路走過的所有地方，感謝一路上所遇到的事物，更加感謝一路上陪著我一起前進的每一個人。

最後，此篇論文獻給學弟妹與在學術界努力前行的各位，祝福大家研究順利！

湯維豪 謹誌

國立臺東大學資訊管理學系碩士班

中華民國一零九年六月

具反雙曲正弦函數特性之不完美除錯 軟體可靠度成長模型

作者：湯維豪

國立台東大學資訊管理學系

摘要

軟體是現代生活中不可或缺的工具，隨著資訊科技的突飛猛進，軟體廣泛的存在於我們的生活周遭，無論是電腦或行動裝置中的軟體，都在生活中扮演著重要的角色；若軟體發生故障，可能造成重大的損失以及帶來不便，因此，軟體的可靠度特別受到重視，為保證軟體的運行正常，已有許多學者提出了各種不同的軟體可靠度成長模型，這些模型被用來預測軟體測試過程中所產生的失效行為，希望藉此提高軟體的可靠度。

而本研究以非齊次卜瓦松過程為基礎，建立一個考慮不完美除錯與學習效果，並具備反雙曲正弦函數特性的軟體可靠度成長模型；藉由七組實際失效數據進行測試，並利用五項模型評估指標將測試結果與十組現有的軟體可靠度成長模型進行比較，結果證實本研究所提之模型具有良好的預測能力，並且於絕大多數情況下，本研究所提之模型的預測能力優於大部分現有模型。

關鍵字：不完美除錯、非齊次卜瓦松過程、軟體可靠度成長模型、反雙曲正弦函數

Imperfect Debugging Software Reliability Growth Model with Inverse Hyperbolic Sine Function

Wei-Hao Tang

Abstract

Software is an indispensable tool in modern life. With the rapid development of information technology, software widely exists around our lives. Whether software in computers or applications in mobile devices, they play an important role in life; if software failures may cause significant losses and inconveniences. Therefore, the reliability of the software is particularly valued. To ensure the status of software is normal, many scholars had proposed a variety of different software reliability growth models. These models had been used to predict the failure behaviors that were found during software testing, hoping to improve the reliability of the software.

Based on the non-homogeneous Poisson process, this study established a software reliability growth model that considered imperfect debugging and learning effects, and has the characteristics of inverse hyperbolic sine function; By running seven sets of actual failure data for testing, the five model evaluation indicators were used to compare the test results with the ten existing software reliability growth models. The results confirmed that the model mentioned in this study has the good predictive ability, and in most cases, the predictive ability this study mentioned of the model is better than the existing models.

Keywords: *Imperfect Debugging; Non-Homogeneous Poisson Process; Software Reliability Growth Models; Inverse Hyperbolic Sine Function*

目錄

謝誌.....	i
摘要.....	ii
Abstract.....	iii
目錄.....	iv
表目錄.....	vi
圖目錄.....	vii
第一章 緒論.....	1
第一節 研究背景與動機.....	1
第二節 研究目的與範圍.....	1
第三節 研究方法與流程.....	2
第二章 文獻探討.....	4
第一節 可靠度.....	4
第二節 失效.....	6
第三節 硬體可靠度與軟體可靠度.....	7
第四節 軟體可靠度成長模型.....	11
第五節 非齊次卜瓦松軟體可靠度成長模型.....	12
第六節 不完美除錯的 NHPP 軟體可靠度成長模型.....	15
第七節 具備學習效果的 NHPP 軟體可靠度成長模型.....	17
第八節 現有的 NHPP 軟體可靠度成長模型.....	21
第三章 研究方法.....	28
第一節 研究架構.....	28
第二節 研究假設.....	29
第三節 函數導入.....	31
第四節 賦予學習效果.....	33
第五節 資料來源.....	38
第六節 參數估計方法.....	38
第七節 模型評估標準.....	40
第八節 模型建構.....	42

第四章	模型測試與資料分析.....	45
第一節	參數估計.....	45
第二節	實際失效數據集.....	48
第三節	模型比較與測試.....	54
第四節	小結.....	96
第五章	結論與建議.....	98
第一節	結論.....	98
第二節	研究限制.....	98
第三節	建議.....	99
參考文獻.....		101
中文部分.....		101
外文部分.....		103
附錄.....		108
附錄一：故障內容函數之限制條件推導過程.....		108

表目錄

表 1 NHPP 軟體可靠度模型及均值函數表	21
表 2 研究假設中的變數說明.....	29
表 3 函數導入中的變數說明.....	31
表 4 賦予學習效果中的變數說明.....	33
表 5 失效數據來源.....	38
表 6 模型評估標準中的變數說明.....	40
表 7 模型建構中的變數說明.....	42
表 8 在線資料紀錄軟體測試資料.....	48
表 9 即時指揮與控制系統.....	48
表 10 Tandem Computers 失效數據	49
表 11 無線網路交換中心失效數據.....	49
表 12 即時控制的應用程式失效數據.....	50
表 13 即時監視與控制系統失效數據.....	51
表 14 鐵路聯鎖系統失效數據.....	52
表 15 資料集一之各模型參數估計數值.....	54
表 16 各模型於資料集一的比較結果.....	60
表 17 資料集二之各模型參數估計數值.....	60
表 18 各模型於資料集二的比較結果.....	66
表 19 資料集三之各模型參數估計數值.....	66
表 20 各模型於資料集三的比較結果.....	72
表 21 資料集四之各模型參數估計數值.....	72
表 22 各模型於資料集四的比較結果.....	78
表 23 資料集五之各模型參數估計數值.....	78
表 24 各模型於資料集五的比較結果.....	84
表 25 資料集六之各模型參數估計數值.....	84
表 26 各模型於資料集六的比較結果.....	90
表 27 資料集七之各模型參數估計數值.....	90
表 28 各模型於資料集七的比較結果.....	96
表 29 本研究所提模型於各組數據之評估標準綜合表現排名表.....	96

圖目錄

圖 1 研究流程圖.....	3
圖 2 硬體失效率與時間之關係.....	8
圖 3 軟體失效率與時間之關係.....	9
圖 4 兩種型態的不完美除錯現象.....	16
圖 5 正加速變化.....	18
圖 6 負加速變化.....	18
圖 7 正負加速變化.....	19
圖 8 平台現象.....	19
圖 9 研究架構圖.....	28
圖 10 資料集一與 G-O Model 之配適度曲線.....	55
圖 11 資料集一與 DSS Model 之配適度曲線.....	55
圖 12 資料集一與 ISS Model 之配適度曲線.....	55
圖 13 資料集一與 Y-LID Model 之配適度曲線.....	56
圖 14 資料集一與 Y-EID Model 之配適度曲線.....	56
圖 15 資料集一與 H/D G-O Model 之配適度曲線.....	56
圖 16 資料集一與 P-Z Model 之配適度曲線.....	57
圖 17 資料集一與 P-N-Z Model 之配適度曲線.....	57
圖 18 資料集一與 P-EID Model 之配適度曲線.....	57
圖 19 資料集一與 Roy Model 之配適度曲線.....	58
圖 20 資料集一與 Arsinh Model 之配適度曲線.....	58
圖 21 資料集一與各模型之配適度曲線比較.....	59
圖 22 資料集二與 G-O Model 之配適度曲線.....	61
圖 23 資料集二與 DSS Model 之配適度曲線.....	61
圖 24 資料集二與 ISS Model 之配適度曲線.....	61
圖 25 資料集二與 Y-LID Model 之配適度曲線.....	62
圖 26 資料集二與 Y-EID Model 之配適度曲線.....	62
圖 27 資料集二與 H/D G-O Model 之配適度曲線.....	62
圖 28 資料集二與 P-Z Model 之配適度曲線.....	63
圖 29 資料集二與 P-N-Z Model 之配適度曲線.....	63
圖 30 資料集二與 P-EID Model 之配適度曲線.....	63
圖 31 資料集二與 Roy Model 之配適度曲線.....	64
圖 32 資料集二與 Arsinh Model 之配適度曲線.....	64
圖 33 資料集二與各模型之配適度曲線比較.....	65
圖 34 資料集三與 G-O Model 之配適度曲線.....	67
圖 35 資料集三與 DSS Model 之配適度曲線.....	67
圖 36 資料集三與 ISS Model 之配適度曲線.....	67

圖 37	資料集三與 Y-LID Model 之配適度曲線.....	68
圖 38	資料集三與 Y-EID Model 之配適度曲線.....	68
圖 39	資料集三與 H/D G-O Model 之配適度曲線.....	68
圖 40	資料集三與 P-Z Model 之配適度曲線.....	69
圖 41	資料集三與 P-N-Z Model 之配適度曲線.....	69
圖 42	資料集三與 P-EID Model 之配適度曲線.....	69
圖 43	資料集三與 Roy Model 之配適度曲線.....	70
圖 44	資料集三與 Arsinh Model 之配適度曲線.....	70
圖 45	資料集三與各模型之配適度曲線比較.....	71
圖 46	資料集四與 G-O Model 之配適度曲線.....	73
圖 47	資料集四與 DSS Model 之配適度曲線.....	73
圖 48	資料集四與 ISS Model 之配適度曲線.....	73
圖 49	資料集四與 Y-LID Model 之配適度曲線.....	74
圖 50	資料集四與 Y-EID Model 之配適度曲線.....	74
圖 51	資料集四與 H/D G-O Model 之配適度曲線.....	74
圖 52	資料集四與 P-Z Model 之配適度曲線.....	75
圖 53	資料集四與 P-N-Z Model 之配適度曲線.....	75
圖 54	資料集四與 P-EID Model 之配適度曲線.....	75
圖 55	資料集四與 Roy Model 之配適度曲線.....	76
圖 56	資料集四與 Arsinh Model 之配適度曲線.....	76
圖 57	資料集四與各模型之配適度曲線比較.....	77
圖 58	資料集五與 G-O Model 之配適度曲線.....	79
圖 59	資料集五與 DSS Model 之配適度曲線.....	79
圖 60	資料集五與 ISS Model 之配適度曲線.....	79
圖 61	資料集五與 Y-LID Model 之配適度曲線.....	80
圖 62	資料集五與 Y-EID Model 之配適度曲線.....	80
圖 63	資料集五與 H/D G-O Model 之配適度曲線.....	80
圖 64	資料集五與 P-Z Model 之配適度曲線.....	81
圖 65	資料集五與 P-N-Z Model 之配適度曲線.....	81
圖 66	資料集五與 P-EID Model 之配適度曲線.....	81
圖 67	資料集五與 Roy Model 之配適度曲線.....	82
圖 68	資料集五與 Arsinh Model 之配適度曲線.....	82
圖 69	資料集五與各模型之配適度曲線比較.....	83
圖 70	資料集六與 G-O Model 之配適度曲線.....	85
圖 71	資料集六與 DSS Model 之配適度曲線.....	85
圖 72	資料集六與 ISS Model 之配適度曲線.....	85
圖 73	資料集六與 Y-LID Model 之配適度曲線.....	86
圖 74	資料集六與 Y-EID Model 之配適度曲線.....	86

圖 75	資料集六與 H/D G-O Model 之配適度曲線	86
圖 76	資料集六與 P-Z Model 之配適度曲線	87
圖 77	資料集六與 P-N-Z Model 之配適度曲線	87
圖 78	資料集六與 P-EID Model 之配適度曲線	87
圖 79	資料集六與 Roy Model 之配適度曲線	88
圖 80	資料集六與 Arsinh Model 之配適度曲線	88
圖 81	資料集六與各模型之配適度曲線比較	89
圖 82	資料集七與 G-O Model 之配適度曲線	91
圖 83	資料集七與 DSS Model 之配適度曲線	91
圖 84	資料集七與 ISS Model 之配適度曲線	91
圖 85	資料集七與 Y-LID Model 之配適度曲線	92
圖 86	資料集七與 Y-EID Model 之配適度曲線	92
圖 87	資料集七與 H/D G-O Model 之配適度曲線	92
圖 88	資料集七與 P-Z Model 之配適度曲線	93
圖 89	資料集七與 P-N-Z Model 之配適度曲線	93
圖 90	資料集七與 P-EID Model 之配適度曲線	93
圖 91	資料集七與 Roy Model 之配適度曲線	94
圖 92	資料集七與 Arsinh Model 之配適度曲線	94
圖 93	資料集七與各模型之配適度曲線比較	95



第一章 緒論

第一節 研究背景與動機

一、研究背景

隨著資訊技術的日新月異，各式各樣的軟體大量出現在我們的生活周遭，軟體的有效運用可以使人們过上更便利的生活，無論是電腦上的應用程式，亦或是手機中的 APP，這些軟體都已成為我們日常生活中不可或缺的重要工具；因此，軟體可靠度(Software Reliability)格外受到重視，為確保軟體的運作能夠正常，過去有許學者提出了各種不同的軟體可靠度成長模型(Software Reliability Growth Models, SRGMs)，這些模型用來預測軟體在測試過程中所產生的失效行為，希望藉由分析軟體的故障發生來提升軟體的品質。

二、研究動機

由於開發一套完全不會發生錯誤的軟體是相當困難的，一旦軟體發生無預期的故障，將會造成使用者及軟體開發商的損失，因此精確的預測錯誤發生是重要的；故本研究將藉由蒐集並研讀文獻，嘗試納入數學函數概念，期望建立一個比現有模型更加精確的軟體可靠度成長模型，並可供於軟體可靠度評估之參考依據。

第二節 研究目的與範圍

一、研究目的

本研究主要期望藉由建構預測能力更為準確的軟體可靠度成長模型，有效預測錯誤的發生，藉此提高軟體的可靠度；並期望本研究所提的想法及概念對於未來的其他研究有所幫助。

二、研究範圍

本研究主要探討在不完美除錯(Imperfect Debugging)情況下，以非齊次卜瓦松過程(Non-Homogeneous Poisson Process, NHPP)為基礎建立的軟體可靠度成長模型，參考各種不同具有趨勢的數學函數進行發想，建構一個新形

態的軟體可靠度成長模型，期望透過數學函數的運用，提升模型的配適度，準確預測錯誤的發生，以此提高軟體的可靠度。

第三節 研究方法與流程

一、研究方法

本研究經過文獻的探討，提出以非齊次卜瓦松過程為基礎，結合函數特性於故障內容函數中，以描寫不完美除錯的過程，並考慮在錯誤偵測具有學習效果的軟體可靠度成長模型。於模型建構完成後，利用七組實際故障資料集，以最小平方估計法（Least Squares Estimation, LSE）求取模型的參數，並繪製出配適度曲線圖，以五個模型評估標準來與過去學者所提出的模型進行比較，藉此判斷本研究所提出的新模型是否比現有模型擁有更好的預測能力。

二、研究流程

本研究之流程首先確立研究問題、動機及目的，然後訂定研究範圍及初步的解決概念，接著開始回顧相關文獻，並記錄想法，然後利用文獻回顧的過程於文獻中蒐集公開的真實失效數據集；在經過充分的閱讀整理以及理解基礎知識之後，本研究採用 NHPP 為基礎，並嘗試導入數學函數用以優化模型，若函數趨勢不符合或是運算困難，則重新尋找；找到合適的函數之後，調整其值域並加入故障內容函數中以建立模型，然後利用微分方程賦予模型學習效果；模型建構完畢之後，進行參數估計，若配適度不佳或是出現不合理的數值時，則針對參數進行適度地微調；參數調整完畢後，進行模型的整體評估，並與以往學者所提模型進行比較，若所提模型的表現普遍較現有模型拙劣，則將其捨棄，然後重新尋找函數用以重新建構模型；獲得良好的模型之後，本研究將彙整結果並針對未來研究方向提出建議。本研究流程圖如圖 1 所示。

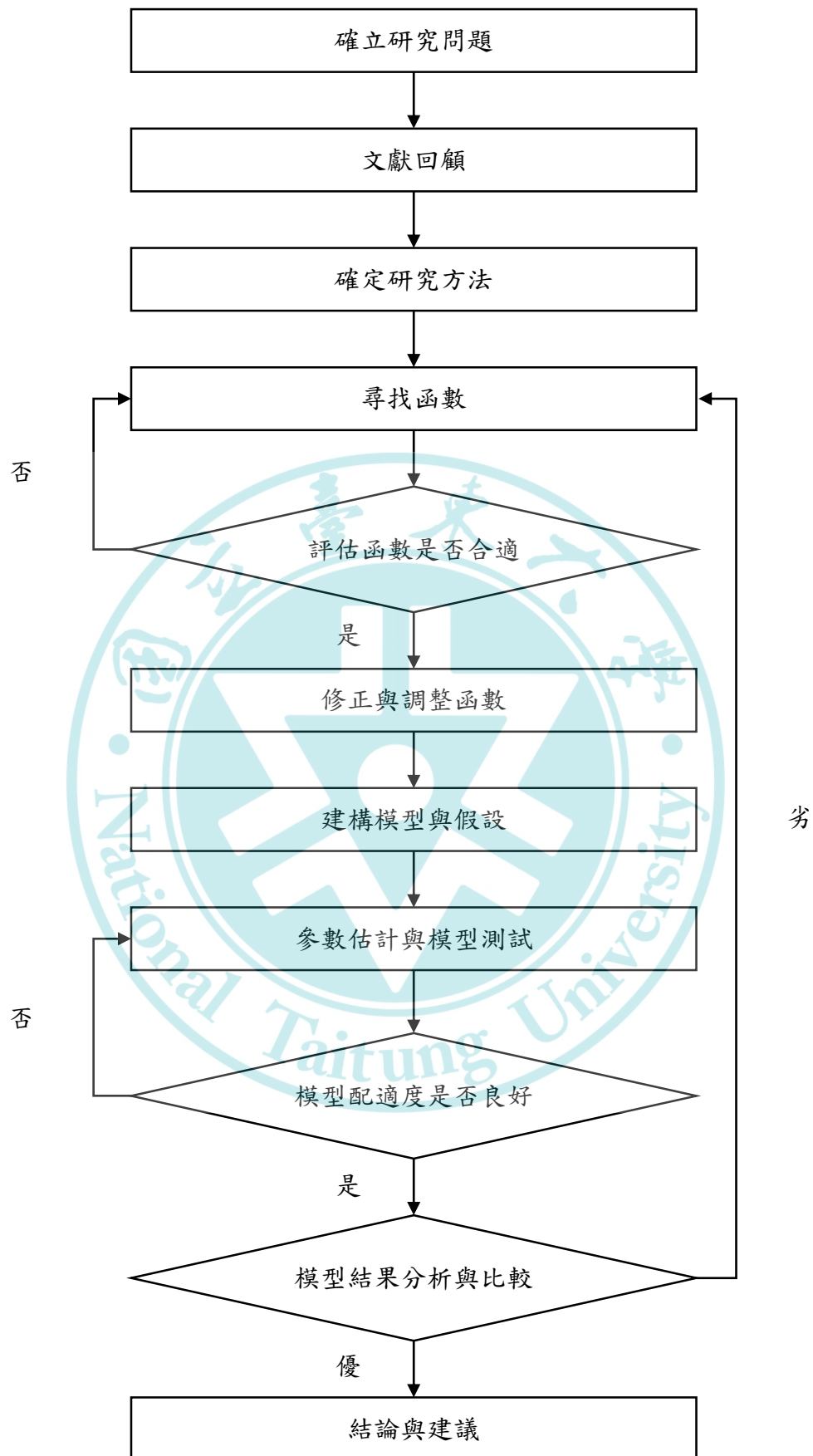


圖 1 研究流程圖

第二章 文獻探討

本研究藉由回顧相關文獻，汲取以往學者的經驗和研究貢獻，經過彙整與發想，並建構一個新的軟體可靠度成長模型，期望能夠更有效且準確的預估錯誤的發生與時間之間的關係，以利於輔助決策之參考。

第一節 可靠度

一、可靠度基本描述

可靠度(Reliability)之定義為：「物品於既定的時間內，在特定的使用(環境)條件下，執行特定性能或功能，圓滿成功達成任務的機率」。其中重要組成要素包含「功能」、「條件」、「時間」和「能力或機率」(彭鴻霖，2000)：

1. 功能：功能(Function)又稱為性能或機能，是物品研發與生產最重要的目的。而物品失去功能的狀態又稱為失效(Failure)，至於物品的功能是否正常則需視物品的設計而定。
2. 條件：條件包含環境條件、工作條件以及維修(Maintenance)條件。環境條件是指溫度、濕度、振動、電磁干擾…等，實際狀態下的環境；工作條件是指物品在運行時的控制條件和動力條件；而環境條件和工作條件的訂定需要根據物品運作的生命週期(Life Cycle)來決定，經過環境輪廓分析訂定其運作所需的標準和細部項目。維護條件則包括定期保養、預防性維護、改正維護、檢測等預防及維護措施所訂定的各類條件。
3. 時間：時間是可靠度的四個要素中最重要，其不單只指花費的時間，它也能包括操作次數、距離、反覆次數等概念。時間的定義需要經由物品的生命週期來決定；實際上，物品依據執行任務功能的初始條件又可分為工作時間(Duty Time)和年紀(Age)兩項；工作時間是指物品實際運行的時間，而年紀則是指物品在時間方面的起始條件。
4. 能力或機率：機率(Probability)或能力(Ability)是可靠度的總體評估標準。能力是指物品執行任務所需的功能，其定義較為質化且廣義；而機率則是針對此能力的量化標準。

二、可靠度相關文獻

針對汽車機電系統設計一套可靠度評估標準，並利用國內某機構設計的馬達驅動控制器系統為例，評估其是否符合 ISO 26262 功能安全需求(Functional Safety Requirement)(王育浚，2017)。

由地震危害度分析著手，以條件均值譜 (Conditional Mean Spectrum) 作為輸入，考慮近斷層地震對結構物的危害，考究在特定的劣化程度與地動強度下橋梁的結構易損性曲線，並提供一套方法來評估在橋梁生命週期內的安全性及可靠度(王若君，2018)。

以網路可靠度(Network Reliability)做為物流績效指標用以評估時窗、時間限制下之多階狀態多產品物流網路。針對各供應商所提供之特定情境提出演算法，求取滿足區域需求及時間限制之網路可靠度，並透過實際案例測試所提之演算法。為更接近真實狀況，考慮各供應商所提供多樣化產品的狀況，並發展新網路可靠度演算法加以計算。透過此績效指標了解多階狀態多產品物流網路之能力，並進行更進一步的決策與分析(莊滿如，2018)。

提出一套以可靠度為基礎的方法，用以分析橋梁的耐洪能力，並用此方法實際分析大甲溪東勢大橋的耐洪能力(賈奕騰，2014)。

以電阻式記憶體中的故障行為進行區分，利用致命細胞取代技術及修正餘度(Correction Slack)的概念提出一套技術，並對一個 64MB 的電阻式記憶體進行修復率、良率、可靠度與硬體成本分析，證實此技術相較於記憶體配置修正能力為 3 的 BCH 碼，修復率最高可提升約 86%，當原始良率為 0.85 時，最低的有效良率能維持在 99.7%，可靠度於 310,000 小時仍可保持在 97%以上，並且硬體成本增加不到 0.3%(謝協成，2020)。

由上述文獻可知，可靠度在各個領域內都佔有重要的地位，各方學者也根據不同領域的需求，提出了許多關於可靠度的相關研究。而本研究則針對軟體可靠度方面進行鑽研，期望對於研究領域有所貢獻。

第二節 失效

與可靠度息息相關的就是失效情形的發生，失效是指物品於實際運用時發生失敗造成錯誤，進而導致物品不能夠於特定時間、環境等條件下完成任務(林哲宇，2019)；當失效頻頻發生，將會大幅影響物品的可靠度。

一、硬體失效

硬體失效(Hardware Failure)的定義為產品、產品的配件或產品的結構發生任何形狀、尺度或材料性質的改變，致使產品無法順利完成其指定功能的情況(彭鴻霖，2000)。依據硬體失效的持續時間，可將其分為暫時性失效與永久性失效兩種(Teng et al., 2006；謝濱壕，2019)：

1. 暫時性失效：是一種暫時性的硬體功能失效狀態，它是由外在環境的干擾所造成的，通常可藉由重新操作而恢復功能。
2. 永久性失效：是一種無法藉由簡易的重新操作而恢復功能的失效狀態，它是由硬體退化與失效所造成的，通常只能藉由更換零件來恢復功能。當硬體零件因退化而停止正常運作之後，該零件將被視為永久性失效。而永久性硬體失效依據影響範圍又可分為總體硬體失效與局部硬體失效兩種：
 - i. 總體硬體失效：整個硬體全面停止運作。
 - ii. 局部硬體失效：部分硬體零件停止運作。

二、軟體失效

軟體失效(Software Failure)是指「產品執行所需功能的能力終止或無法在先前指定的範圍內執行功能」；又或者是「系統或系統組件發生在指定範圍內未執行所需功能的事件」。

軟體故障(Software Fault)可能會產生軟體失效(ISO/IEC/IEEE 24765, 2010)。

而軟體失效可分為硬失效(Hard Failure)與軟失效(Soft Failure)兩種(ISO/IEC/IEEE 24765, 2010)：

1. 硬失效：導致系統完全關閉的失效。
2. 軟失效：允許系統保有部分操作功能持續運作的失效。

三、硬體失效及軟體失效相關文獻

分析用於數位微電子裝置的銅鋁合金材料於不同測試環境下的可靠度及失效情況，以確認安裝在汽車或航空硬體設備上的電子元件系統具備較高的可靠度(劉建邦，2019)。

運用非齊次卜瓦松過程，並考慮多種失效情境及不完美除錯的情況，建構軟體可靠度成長模型與成本分析函數，在考量軟體檢測成本及可接受之軟體可靠度標準之下，求取最佳軟體上市時間，並進一步對此實例之成本函數進行敏感度分析(林邑築，2018)。

透過文獻回顧與彙整，提出一些建議以增加可靠度預測模型的可用性，此些建議或許有助於開發人員減少失效率，並增強軟件可靠度(Sahu & Srivastava, 2020)。

提出了一種非參數方法來進行元件的生存分析，藉由失效數據可計算出軟體的失效率和可靠度，其中失效率隨時間增加，而可靠度隨時間減少(Chopra, Nautiyal, Malik, Ram & Sharma, 2020)。

失效當然是我們不樂見的事情，失效不但會造成財物的損失，嚴重甚至導致人員傷亡，如：汽車剎車失靈導致事故又或是導行失靈導致用路人墜谷…等；造成失效的可能原因有很多，因此，進行失效相關的研究有助於我們了解及避免失效的發生，本研究就針對軟體方面的失效進行鑽研，期望藉由研發軟體可靠度模型，提前預知失效的發生，以利於降低風險。

第三節 硬體可靠度與軟體可靠度

另外，在電子資訊相關領域之中可靠度主要可分為軟體可靠度與硬體可靠度(Hardware Reliability)。相較於軟體可靠度，硬體可靠度的發展較早，因此具備較成熟的評估技術，其主要著重於解析失效數據。而軟體可靠度發展相對較晚，其主要著重於模型的建構與參數的解釋(李嘉華，2014)。

一、硬體可靠度

硬體可靠度是指在特定的條件下，硬體於規定時間內運作，並且沒有發生故障的機率。硬體可靠模型是定量的且與時間有關。而可靠度理論中，浴缸曲線(Bathtub Curve)較適合描述硬體的壽命週期，如圖 2 所示；基本上浴缸曲線可分成三個時間區段，關於浴缸曲線的描述如下(Poong, 2009)：

1. 早期失效時間區段(Early Failure Period Region)：在早期失效時間區段之中，失效主要發生在設計、製造、處理及安裝過程中，其失效發生的主要原因歸咎於本身材料的缺陷。隨著時間推進，此區段的失效率將隨之降低。早期失效的時間區段可以是幾個星期到幾個月。
2. 隨機失效時間區段(Random Failure Period Region)：在隨機失效時間區段之中，失效主要發生的原因歸咎於不穩定的環境和意外事故。此區段的失效率是相對恆定的，且其占硬體的整個生命週期中的大部分時間。
3. 磨損時間區段(Wear-out Period Region)：在磨損的時間區段之中，失效主要發生的原因歸咎於材料的老化或其他機制。

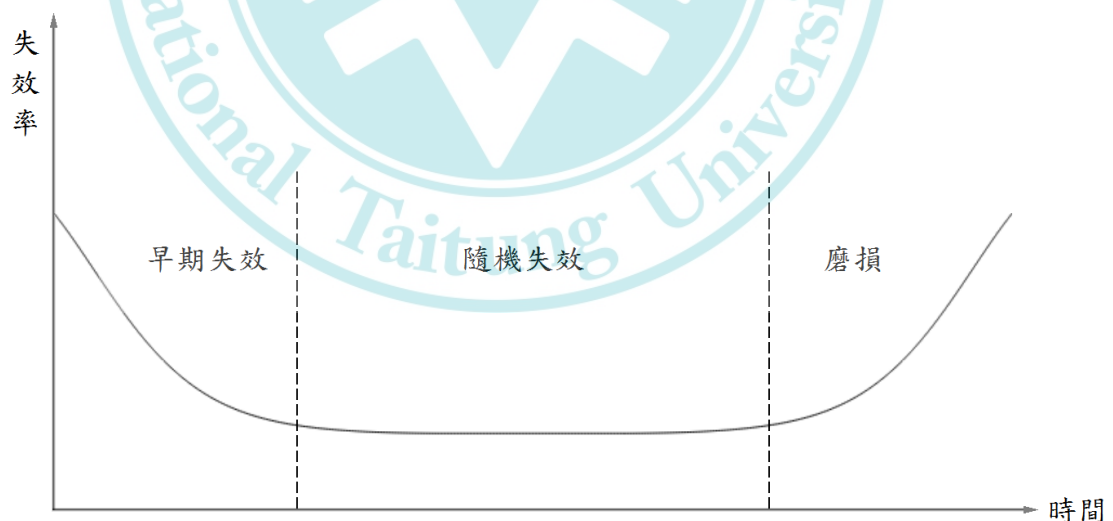


圖 2 硬體失效率與時間之關係

資料來源：(Poong, 2009)

二、軟體可靠度

相較於硬體可靠度，軟體可靠度的發展比較晚，其基礎概念大多由硬體可靠度發展而來，因此，軟體可靠度的定義與硬體可靠度相似，即是指在特定的條件下，軟體於規定時間內成功運行，並且沒有發生故障的機率 (Lyu, 1996； 江文馨，2013)。

不過軟體可靠度與硬體可靠度也有不同之處，其不同之處便在於硬體隨著使用時間增加，在磨損時間區段，失效的情況將會增加；其主要原因是因為硬體設備會隨著使用時間而老化、磨損，最終造成可靠度下降；而軟體並不會老化、磨損，軟體的失效原因大多由程式碼的錯誤所產生，如邏輯錯誤、資料錯誤等等因素而產生。雖然軟體可藉由更新取得更多更好的性能，但隨著軟體的複雜度上升，發生失效的機率也將因此而提升(林哲宇，2019)。軟體失效率與時間的關係如圖 3 所示。



圖 3 軟體失效率與時間之關係

資料來源：(Kapur, Pham, Gupta & Jha, 2011b)

三、硬體可靠度及軟體可靠度相關文獻

提出整合錯誤修正碼和硬體修復用以提升嵌入式記憶體良率及可靠度(ECC-enhanced BISR, EBISR)的技術，該技術能夠在系統運作時保持原本錯誤修正碼的可靠度(蔡政儒，2015)。

以影像處理電路為例，配合影像處理及其他電路設計與測試的觀念提出一套可以提升影像處理電路良率及可靠度的硬體設計架構(顧家齊，2013)。

蒐集 1986 年至 2018 年之間關於軟體可靠度領域相關的文章，選取了其中 54 篇文章，依據解決方案進行分類，並進行分析和比較，用以預測不斷發展的系統可靠性(Alhazzaa & Andrews, 2019)。

針對 SonarQube 軟體系統進行一系列涉及不同領域和技術的實驗，以驗證對於在 SonarQube 中已知可靠度參數的感知影響(Perceived Influence)，透過這些實驗以及數據分析，確定了影響軟體可靠度的一系列影響因素(Joshi, Deshpande & Punnekkat, 2019)。

以物理失效模式(Physics of Failure, PoF)為基礎設計一套電腦系統，用以模擬產品建模、負載應力分析與可靠度分析，再利用一個實際案例進行研究。結果證明，此系統可以有效提高可靠度，且確保精度，具備高計算效率的分析(Sun et al., 2020)。

藉由觀察文獻可以發現，在初期測試階段，軟體可靠度與硬體可靠度的失效趨勢相近，隨著軟體的測試及修正亦或是硬體的預燒(Burn in)，失效率逐漸的下降；而在中期階段，也就是軟體與硬體的主要使用階段，雖然失效率同樣的普遍低於初期測試階段，屬於較穩定的階段；但兩者相比較起來，軟體雖然能夠藉由更新或升級取得擴充的功能，但失效率的穩定度則產生變動，因此，相較於硬體可靠度，軟體的更新或升級將導致軟體可靠度的穩定性下降；而到了後期階段，硬體逐漸因為磨損以及老化而導致可靠度下降，最終硬體也將面臨報廢或汰換的命運；而軟體不會老化或磨損，因此相較於硬體可靠度，軟體可靠度在後期階段不會有太大變化，但軟體會因為各種原因而漸漸過時，隨著各類軟體的推陳出新，老舊的軟體被使用的頻率也漸漸下降，當軟體帶來的效益不足以維持自身升級與

維護所需的成本時，該軟體將停止營運或不再更新，並漸漸的被人們遺忘。

無論是硬體可靠度亦或是軟體可靠度，在電子資訊相關領域中都佔有非常重要地位，國內外也有許多學者提出各種不同的論文及觀點，而本研究則側重於觀察與分析軟體可靠度方面的問題。

第四節 軟體可靠度成長模型

一、軟體可靠度模型基本描述

軟體可靠度模型(Software Reliability Model)可分為缺陷密度(Defect Density)模型及軟體可靠度成長模型兩種，描述如下(Wood, 1996；李嘉華，2014；謝濱壕，2019)：

1. 缺陷密度模型：是一種藉由分析軟體程式碼的行數、程式特性、輸入資料以及輸出資料等設計參數來預測軟體中的錯誤數目，用以分析評估軟體的可靠度。
2. 軟體可靠度成長模型：是一種藉由實際失效數據做為測試資料結合某些具備特定趨勢的函數來預測軟體運行時的失效情形，從而評估軟體的可靠度。

Pham 則把失效發生與錯誤移除等因素以時間機率區分成七種，分別為：植錯型(Error Seeding)、失效率(Failure Rate)、曲線符合(Curve Fitting)、可靠度成長(Reliability Growth)、馬可夫架構(Markov Structure)、時間序列(Time-series)、非齊次卜瓦松過程(Pham, 2006；李嘉華，2014；王亭雅，2015)。

二、軟體可靠度模型相關文獻

提出一個失效數有限的 NHPP 軟體可靠度成長模型，並利用三組實際失效數據集與四項模型評估標準進行驗證，證明所提出的模型比現有的有限失效類別模型表現更佳(Hanagal & Bhalerao, 2019)。

提出一種選擇軟體可靠度成長模型的方法，此方法使用修正的布穀鳥搜索優化法(Cuckoo Search Optimization)以及人工蜂群演算法(Artificial Bee Colony Algorithm)來尋找更有效性的可靠度模型。經過評估可靠性模型的各種度量，

並與其他模型進行比較，可以確定使用這些演算法能有效地找到最佳模型，以利於評估軟體的可靠度(Rao & Anuradha, 2017)。

軟體可靠度這個議題一直受到很大的重視，畢竟現代生活中軟體無所不在，因此，能夠長期且穩定運作的軟體對於現今社會是不可或缺的，許多學者提出了諸多方法來分析軟體的運行及其可靠度，而本研究採用的是以非齊次卜瓦松過程為基礎建立的軟體可靠度成長模型做為主題的研究。

第五節 非齊次卜瓦松軟體可靠度成長模型

一、卜瓦松過程定義

卜瓦松過程包含了獨立性和卜瓦松分布兩個重要概念，其定義為，一個整數的隨機過程 $\{X(t); t \geq 0\}$ ，其強度或機率稱為 λ ，而 λ 必須大於零，其條件描述如下(Taylor & Karlin, 1998)：

1. 對於任何時間點都是增加的過程，表示為 $t_0 = 0 < t_1 < t_2 < \dots < t_n$ ，而在此增加的過程中，變化量表示為 $X(t_1) - X(t_0), X(t_2) - X(t_1), \dots, X(t_n) - X(t_{n-1})$ ，其中的每一個變化量都是獨立且隨機產生的；
2. 對於 $s \geq 0$ 且 $t > 0$ 的情況下，隨機變化量 $X(s+t) - X(s)$ 遵循卜瓦松分配，表示為

$$\Pr\{X(s+t) - X(s) = k\} = \frac{(\lambda t)^k e^{-\lambda t}}{k!} \text{ for } k = 0, 1, \dots; \quad (2.1)$$

3. $X(0) = 0$ ，初始時間為零，變化量的起始值也為零。

尤其是在觀察變化量 $X(t)$ 時，若其為強度 λ 大於零的卜瓦松過程，則此時平均數 $E[X(t)] = \lambda t$ ，且變異數 $\text{Var}[X(t)] = \sigma_{X(t)}^2 = \lambda t$ ，兩者是相等的。

在卜瓦松過程 $X(t)$ 中， λ 代表在任意極小時間區間之內發生事件的機率。為了更清楚地解釋這一點，列出以下公式(Taylor & Karlin, 1998)：

$$\begin{aligned}
\Pr\{X(t+h) - X(s) = 1\} &= \frac{(\lambda h)e^{-\lambda h}}{1!} \\
&= (\lambda h) \left(1 - \lambda h + \frac{1}{2}\lambda^2 h^2 - \dots\right) \\
&= \lambda h + o(h)
\end{aligned} \tag{2.2}$$

其中 $o(h)$ 代表其餘小於 h 且未被列出的一般項。

而在許多相關應用中，會去考慮卜瓦松過程的強度或機率 $\lambda = \lambda(t)$ 的情況，即是指強度或機率將變成一個隨時間變化而產生改變的函數，此情況稱為非齊次卜瓦松過程；因此，我們藉判斷卜瓦松過程的強度或機率函數 $\lambda(t)$ 是否隨時間產生改變來由區分齊次卜瓦松過程與非齊次卜瓦松過程(Taylor & Karlin, 1998)。

若 $X(t)$ 是一個非齊次卜瓦松過程，並具備強度或機率函 $\lambda(t)$ ，則可由一個增量 $\lambda(t) - \lambda(s)$ 中獲得在 $(s, t]$ 時間區間中的事件數量，其為具有參數 $\int_s^t \lambda(u) du$ 的卜瓦松分布，且在不重疊的時間區段內具有獨立且隨機的增量(Taylor & Karlin, 1998)。

二、非齊次卜瓦松軟體可靠度成長模型描述

當計數過程 $\{N(t), t \geq 0\}$ 為非齊次卜瓦松過程時，則它將具有失效強度函數 (Intensity Function) $\lambda(t), t \geq 0$ ，且 $N(t)$ 呈現卜瓦松分布，並具有均值函數 (Mean Value Function, MVF) $m(t)$ ，以公式表達如下(Pham & Zhang, 1997)：

$$\Pr\{N(t) = k\} = \frac{[m(t)]^k}{k!} e^{-m(t)}, \quad k = 0, 1, 2, \dots, \tag{2.3}$$

根據定義，均值函數 $m(t)$ 是測試所發現故障次數的累積，可以用失效強度函數來表達，公式如下(Pham & Zhang, 1997)：

$$m(t) = \int_{t_0}^t \lambda(s) ds \tag{2.4}$$

其中 $\lambda(s)$ 稱為失效強度函數或失效率(Failure Rate)函數。

NHPP 模型的最主要問題是確定適當的均值函數，用以預測特定時間點所發現的故障數量。

軟體可靠度 $R(x/t)$ 定義為在時間 $(t, t+x)$ 之間沒有發生故障的機率，假設最後一次發生故障的測試時間點為 t ，則軟體可靠度 $R(x/t)$ 表示為以下公式：

$$R(x/t) = e^{-[m(t+x)-m(t)]} \quad (2.5)$$

其中包含某些特殊情況，例如，當 $t = 0$ ，則 $R(x/t) = e^{-m(x)}$ ；當 $t = \infty$ ，則 $R(x/t) = 1$ 。

在廣義的 NHPP 軟體可靠度模型中，均值函數 $m(t)$ 可藉由求解以下微分方程獲得(Pham & Zhang, 1997)：

$$\frac{dm(t)}{dt} = b(t)[a(t) - m(t)], \text{ with } m(t_0) = m_0 \quad (2.6)$$

其中 $a(t)$ 為時間 t 時的總故障內容(Total Fault Content)， $b(t)$ 為錯誤偵測率(Error Detection Rate)，通過此方法，我們可以藉由假設不同的 $a(t)$ 與 $b(t)$ ，從而獲得不同 NHPP 軟體可靠度模型的均值函數 $m(t)$ 。

三、非齊次卜瓦松軟體可靠度成長模型相關文獻

將隨著時間變化而產生變化的故障偵測率概念整合至包含 S 型函數的軟體可靠度成長模型中。實驗結果證明，所提出的模型有助於提高軟體可靠度評估的準確性(Chatterjee & Shukla, 2015)。

將巴斯擴散測試成效函數(Bass Diffusion Testing-Effort Function)納入 NHPP 軟體可靠度成長模型，並遵循其相關限制條件，以真實數據進行實驗，實驗結果證明，提出的模型具有良好的配適度(Rafi & Akthar, 2011)。

本節主要回顧關於卜瓦松過程與 NHPP 軟體可靠度成長模型建構的方法；卜瓦松過程是一種隨機過程也是計數過程，它被廣泛運用於各個領域(周峰成，2001；葉芳瑜，2019)，而 NHPP 則是為了描述更複雜的現實現象而由卜瓦松過程演變而來。本研究利用 NHPP 為基礎來建構軟體可靠度成長模型，透過回顧過去學者的論文，本研究掌握了求取 NHPP 軟體可靠度成長模型均值函數的核心概念與方法，即是設計良好的故障內容函數(Fault Content Function) $a(t)$ 與錯誤偵測率函數(Error Detection Rate Function) $b(t)$ 。

第六節 不完美除錯的 NHPP 軟體可靠度成長模型

一、軟體可靠度成長模型的除錯描述

除錯(Debug)是指偵測、發現、修正和清除程序中的錯誤(ISO/IEC/IEEE 24765, 2010)。

完美除錯(Perfect Debugging)的假設即是指每當嘗試消除故障時，都會將其完美消除，並且不會產生新的故障。但在大多數現實的情況中，由於測試團隊的能力、測試案例的品質、測試環境、測試效率以及測試所花費的時間等等原因，除錯的過程是不完美的(Kapur et al., 2011b)。因此，不完美除錯的假設即是指在偵測到故障時，無法完美地清除故障，並且原始故障可能會保留或被其他故障替換，甚至排除故障的過程中可能產生新的故障，而不完美除錯的形態可分為兩種：

1. 故障被認為已被完全修復，但事實上它並未被完全修復，而當在同一輸入上進行檢查時，該故障將再次導致相同類型的失效。
2. 故障確實被完全修復，但在排除故障原因的同時又產生了其他的故障。

在考慮不完美除錯的情況下，關於軟體可靠度成長模型的失效與故障排除過程如圖 4 所示(Kapur et al., 2011b； 李嘉華，2014； 王亭雅，2015； 謝濱壕，2019)：

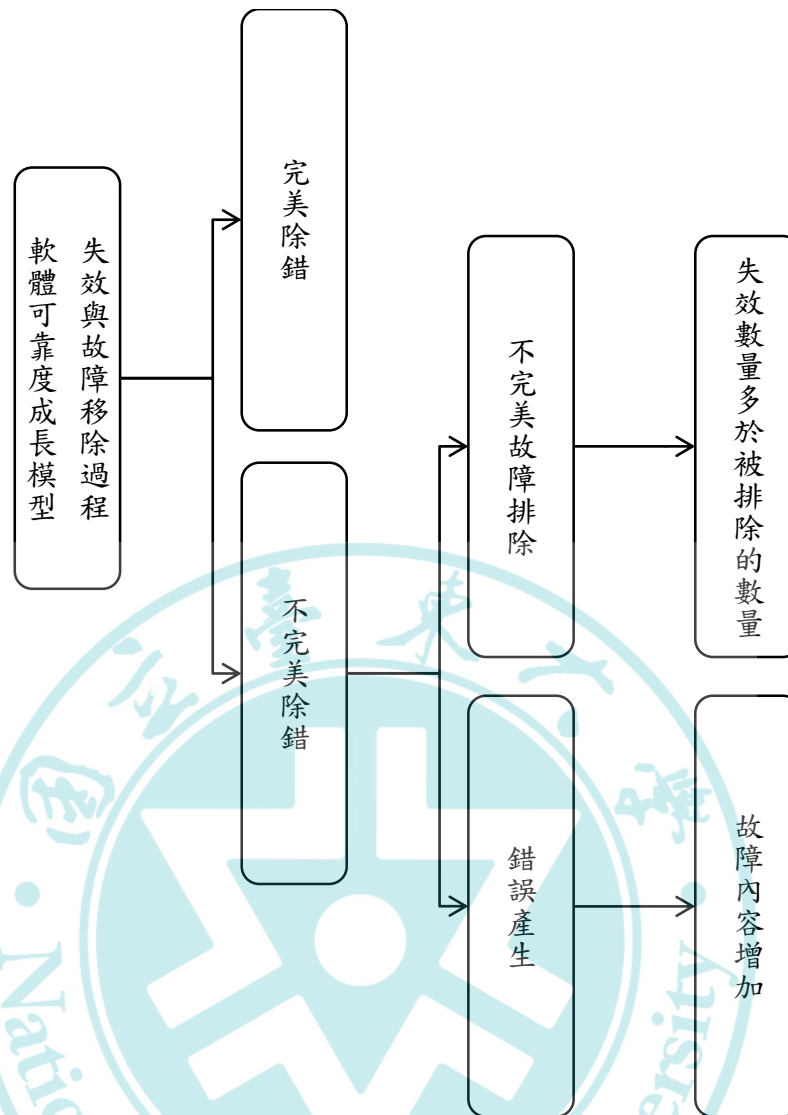


圖 4 兩種型態的不完美除錯現象

資料來源：(Kapur et al., 2011b)

二、不完美除錯的 NHPP 軟體可靠度成長模型相關文獻

提出一個用於多版本軟體開發環境的新概念，並以此概念提出新的不完美除錯 NHPP 軟體可靠度成長模型，此模型使用已在市面上發布，且具有四次版本更新的軟體所得到的實際數據進行測試，並得到驗證(Aggarwal, Kapur & Garmabaki, 2011)。

提出考慮使用者數量以及不完美除錯情況下的 NHPP 軟體可靠度成長模型，以巴斯擴散模型(Bass Diffusion Model)描述使用者數量增加的情況，利用 GNOME 2.0 的失效數據集進行測試，使用四種模型評估標準來計算所提出的模

型的準確性，並證實不完美除錯模型的預測能力優於完美除錯模型(Gandhi, Gondwal & Tandon, 2018)。

完美除錯是指移除故障時，所有故障將會瞬間被完美的移除，且不會引發新的故障，但此種假設並不符合現實中的情況(Yamada, Tokuno & Osaki, 1992)，而關於不完美除錯的現象，許多軟體可靠度成長模型是在故障內容函數中加以描述，若故障內容函數為隨時間增加而遞增的函數，則稱為不完美除錯；因此本研究將考慮不完美除錯的情況，並針對故障內容函數進行設計，期望藉此提升軟體可靠度成長模型的準確率。

第七節 具備學習效果的 NHPP 軟體可靠度成長模型

一、學習效果基本描述

學習效果(Learning Effect)源自於 Wright(1936)的文章，他以學習曲線(Learning Curve)描述生產某些飛機機身的直接人工成本隨著經驗累積而逐漸下降(Wright, 1936；邱桂珍，2008)。有許多研究已證實了此種現象，並在考量學習效果的影響下，將其用於許多應用領域(邱桂珍，2008)。而在軟體可靠度成長模型的研究中也有愈來愈多學者考慮了學習效果對於模型的影響；當測試的時間拉長，測試人員將對偵測與排除錯誤的作業更加熟悉，隨著經驗的累積，測試人員將能有效率的降低軟體中的錯誤、提升偵測與排除錯誤的效率以及提升軟體的可靠度(江文馨，2013)。

在學習的歷程中，學習效果會受到學習者的個人因素以及外在環境因素所干擾，使得學習效果出現不同的變化，以學習時期的先後來說，學習速度將會呈現以下四種不同的變化特性(張琇惟，2008；林哲宇，2019)：

1. 正加速變化：初期學習效果上升緩慢，後期則快速上升，如圖 5 所示。

導致此現象的原因可能是學習者在學習初期缺乏學習動力或是初期較難適應作業內容；而隨著學習動力提升或是漸漸適應工作，因此學習效果於後期有所提升。

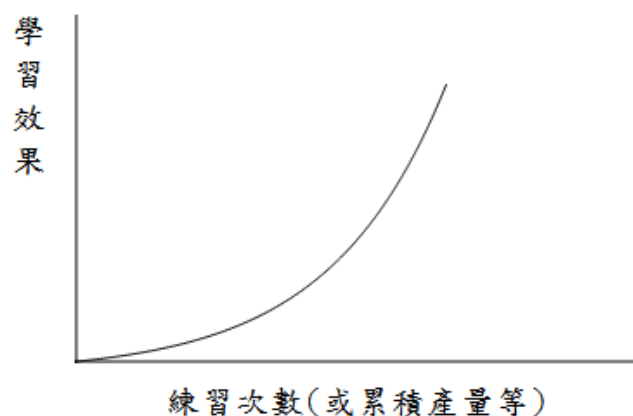


圖 5 正加速變化

資料來源：(2008，張琇惟)

2. 負加速變化：初期學習效果快速提升，後期則漸趨緩慢，如圖 6 所示。導致此現象的原因可能是學習者在學習初期積極學習或是學習者已有類似的學習經驗，亦或許是訓練初期難度較低，因此初期學習效果得以快速提升；而隨著訓練難度的提高，導致後期學習效果逐漸趨緩。



圖 6 負加速變化

資料來源：(2008，張琇惟)

3. 正負加速變化：包含了正加速變化以及負加速變化兩種特性，初期的學習速度由緩慢轉為快速，呈現正加速變化；而後期的學習速度又由快速

轉為緩慢，呈現負加速變化，綜合兩種特性，使得學習曲線呈現 S 型的變化，如圖 7 所示。

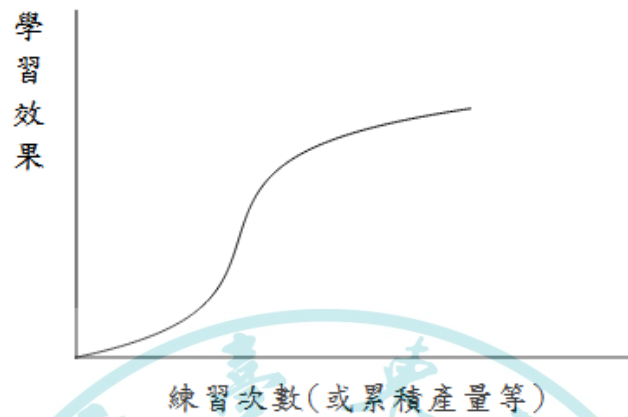


圖 7 正負加速變化

資料來源：(2008，張琇惟)

4. 學習高原：在學習的後期，學習效果趨緩收斂的現象，稱為平台現象，如圖 8 所示。此現象是指學習者學習到一定的程度之後幾乎不再進步，使得學習曲線呈現趨近水平的狀態。

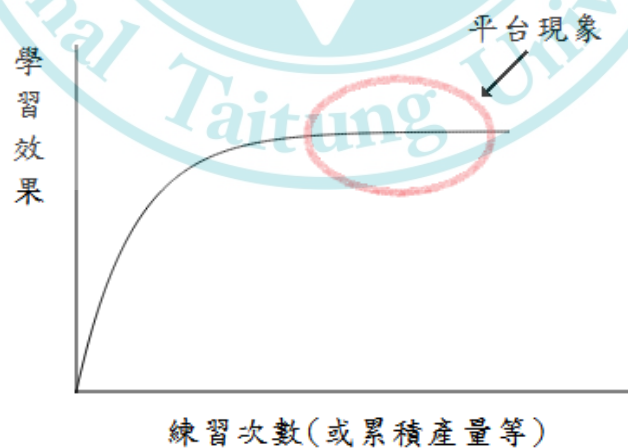


圖 8 平台現象

資料來源：(2008，張琇惟)

二、具備學習效果的 NHPP 軟體可靠度成長模型的相關文獻

提出一種具有可以靈活描述軟體失效及可靠度曲線的 NHPP 軟體可靠度成長模型，考慮測試效率、錯誤偵測率取決於時間(可以描述學習效果)以及故障識別與移除的延遲現象，利用赤池信息量準則(Akaike Information Criterion, AIC)與均方誤差 (Mean of Squared Errors, MSE) 等評估標準與現有模型進行比較，透過不同的實際軟體失效數據集進行驗證，可以證明所提模型具備適用性(Kapur, Goswami, Bardhan & Singh, 2008)。

以學習效果的角度提出一套可以合理的同時描述 S 型及指數型兩種形態表現的模型建構方法，經過分析以及與其他軟體可靠度模型進行比較之後，結果顯示所提模型具有較良好的配適度(Chiu, Huang & Lee, 2008)。

利用 SRGM 中的線性和指數學習效應來增強 Chiu 等人(2008)的模型，並利用多元判定係數(R^2)及四組真實失效數據集與其他模型比較。結果顯示所提的模型的表現優於其他模型，並驗證了學習效果會在軟體測試過程中發生(Chiu, 2013)。

學習效果已在許多領域被運用(邱桂珍，2008)，而在軟體可靠度的相關文獻中，首次提及學習的是說明以 S 型增長曲線描述的軟體錯誤偵測過程可視為測試團隊成員熟悉測試的學習過程(Yamada, Ohba & Osaki, 1983)；而在 Pham、Nordmann 和 Zhang 的模型中提到錯誤偵測率函數為遞增的 S 型曲線，可描述軟體測試人員的學習過程(Pham, Nordmann & Zhang, 1999)。因此，本研究參考 Pham、Nordmann 和 Zhang 所提出的模型，將學習效果納入錯誤偵測率函數的假設中，以錯誤偵測率隨著測試時間增加而增加的現象來描述軟體測試人員學習效果。

第八節 現有的 NHPP 軟體可靠度成長模型

依據文獻，本研究整理出指數型、S 型、完美與不完美除錯與具備學習效果…等，各類不同的 NHPP 軟體可靠度成長模型，如表 1 所示。

表 1 NHPP 軟體可靠度模型及均值函數表

模型名稱(作者,年份)	模型之均值函數
Goel-Okumoto Model (Goel & Okumoto, 1979)	$a(t) = a$ $b(t) = b$ $m(t) = a(1 - e^{-bt})$
Delayed S-shaped Model (Yamada, Ohba & Osaki, 1983)	$a(t) = a$ $b(t) = \frac{b^2 t}{bt+1}$ $m(t) = a(1 - (1 + bt)e^{-bt})$
Inflection S-shaped Model (Ohba, 1984a)	$a(t) = a$ $b(t) = \frac{b}{1+\beta e^{-bt}}$ $m(t) = \frac{a(1-e^{-bt})}{1+\beta e^{-bt}}$
Yamada Linear Imperfect Debugging Model (Yamada, Tokuno & Osaki, 1992)	$a(t) = a(1 + \alpha t)$ $b(t) = b$ $m(t) = a\left(1 - \frac{\alpha}{b}\right)(1 - e^{-bt}) + a\alpha t$
Yamada Exponential Imperfect Debugging Model (Yamada, Tokuno & Osaki, 1992)	$a(t) = ae^{\alpha t}$ $b(t) = b$ $m(t) = \frac{ab}{b+\alpha}(e^{\alpha t} - e^{-bt})$
H/D G-O Model (Hossain & Dahiya, 1993)	$m(t) = \ln\left(\frac{e^a - c}{e^{ae^{-bt}} - c}\right)$
Pham-Zhang Model (Pham & Zhang, 1997)	$a(t) = c + a(1 - e^{-\alpha t})$ $b(t) = \frac{b}{1+\beta e^{-bt}}$ $m(t) = \frac{(a+c)(1-e^{-bt}) - \frac{ab}{b-\alpha}(e^{-\alpha t} - e^{-bt})}{1+\beta e^{-bt}}$
Pham-Nordmann-Zhang Model (Pham, Nordmann & Zhang, 1999)	$a(t) = a(1 + \alpha t)$ $b(t) = \frac{b}{1+\beta e^{-bt}}$ $m(t) = \frac{a\left(1 - \frac{\alpha}{b}\right)(1 - e^{-bt}) + a\alpha t}{1+\beta e^{-bt}}$
Pham Exponential Imperfect Debugging Model (Pham, 2006)	$a(t) = \alpha e^{\beta t}$ $b(t) = \frac{b}{1+ce^{-bt}}$ $m(t) = \frac{ab}{b+\beta} \frac{(e^{\beta t} - e^{-bt})}{1+ce^{-bt}}$
Roy Model (Roy, Mahapatra & Dey, 2014)	$a(t) = a(\alpha - e^{-\beta t})$ $b(t) = b$ $m(t) = a\alpha(1 - e^{-bt}) - \frac{ab}{b-\beta}(e^{-\beta t} - e^{-bt})$

資料來源：本研究整理

一、Goel-Okumoto 模型

Goel-Okumoto 模型(又稱指數 NHPP 模型),本文將此模型簡稱為 G-O Model, G-O Model 是由 Goel 與 Okumoto 兩位學者於 1979 年所提出,經由 Grottke 的整理,該模型的假設如下(Grottke, 2001):

1. 在時間 t 之前所發生的失效遵循具有均值函數 $m(t)$ 的卜瓦松分配。此均值函數具有邊界條件 $m(0) = 0$, 以及 $\lim_{t \rightarrow \infty} m(t) = N < \infty$, 即是指失效數量有限。
2. 在時間區間 $(t, t + \Delta t], \Delta t \rightarrow 0$ 之中,發生失效的數量與軟體內未檢測到的故障數量 $N - m(t)$ 成正比,其中比例常數為 $\emptyset$ 。
3. 對於時間 $t_1 < t_2 < \dots < t_n$ 的任何有限集合,在所有不相交的時間間隔 $(0, t_1), (t_1, t_2), \dots, (t_{n-1}, t_n)$ 中發生故障的數量是相互獨立的。
4. 每當發生失效時,引起失效的故障將立即被移除,並且不會產生新的故障。

以廣義的 NHPP 軟體可靠度模型表示,即假設 $a(t) = a$ 、 $b(t) = b$ 以及 NHPP 基本的假設,初始值 $m(0) = 0$ 等條件,藉由求解微分方程(2.6)可得到:

$$m(t) = a(1 - e^{-bt}) \quad (2.7)$$

二、Delayed S-shaped 模型

Delayed S-shaped 模型,本文將此模型簡稱為 DSS Model, DSS Model 是由 Yamada、Ohba 與 Osaki 三位學者於 1984 年所提出,經由 Hanagal 與 Bhalerao 的整理,該模型的假設如下(Hanagal & Bhalerao, 2018):

1. 由失效偵測的觀點所示,在程序之中所有的故障都是相互獨立的。
2. 在任何時間點的失效偵測率與目前軟體中的故障數量成正比。
3. 失效偵測的比例為常數。
4. 軟體中的初始故障內容是一個隨機變量。

5. 軟體系統會在隨機時間點發生失效，而這些失效是由系統中存在的錯誤所導致。
6. 第 $i-1$ 次失效與第 i 次失效的時間間隔取決於第 $i-1$ 次失效的時間。
7. 每當發生失效時，引起失效的故障將立即被移除，並且不會產生新的故障。

以廣義的 NHPP 軟體可靠度模型表示，即假設 $a(t) = a$ 、 $b(t) = \frac{b^2 t}{bt+1}$ 以及 NHPP 基本的假設，初始值 $m(0) = 0$ 等條件，藉由求解微分方程(2.6)可得到：

$$m(t) = a(1 - (1 + bt)e^{-bt}) \quad (2.8)$$

三、Inflection S-shaped 模型

Inflection S-shaped 模型，本文將此模型簡稱為 ISS Model，ISS Model 是由 Ohba 於 1984 年所提出，該模型的假設如下(Ohba, 1984a)：

1. 錯誤偵測率與目前程序中的錯誤數量成正比；
2. 且在測試(或觀察)的期間內，此比例逐漸增加。
3. 每當軟體發生失效時，引起失效的故障將立即被移除，並且不會產生新的故障。

以廣義的 NHPP 軟體可靠度模型表示，即假設 $a(t) = a$ 、 $b(t) = \frac{b}{1+\beta e^{-bt}}$ 以及 NHPP 基本的假設，初始值 $m(0) = 0$ 等條件，藉由求解微分方程(2.6)可得到：

$$m(t) = \frac{a(1-e^{-bt})}{1+\beta e^{-bt}} \quad (2.9)$$

四、Yamada Linear Imperfect Debugging 模型

Yamada Linear Imperfect Debugging 模型，本文將此模型簡稱為 Y-LID Model，Y-LID Model 是由 Yamada、Tokuno 與 Osaki 三位學者於 1992 年藉由修正 Goel-Okumoto 模型的假設所提出，該模型的假設如下(Yamada, Tokuno & Osaki, 1992)：

1. 軟體系統中的故障將會造成軟體隨機出現失效。
2. 每當發生軟體失效時，引起失效的軟體故障將被移除，但移除故障的過程中可能會產生新的故障。
3. 每單位測試時間偵測到的失效數量與軟體系統中目前的故障內容成正比。

以廣義的 NHPP 軟體可靠度模型表示，即假設 $a(t) = a(1 + \alpha t)$ 、 $b(t) = b$ 以及 NHPP 基本的假設，初始值 $m(0) = 0$ 等條件，藉由求解微分方程(2.6)可得到：

$$m(t) = a \left(1 - \frac{\alpha}{b}\right) (1 - e^{-bt}) + a\alpha t \quad (2.10)$$

五、Yamada Exponential Imperfect Debugging 模型

Yamada Exponential Imperfect Debugging 模型，本文將此模型簡稱為 Y-EID Model，Y-EID Model 是由 Yamada、Tokuno 與 Osaki 三位學者於 1992 年藉由修正 Goel-Okumoto 模型的假設所提出，該模型的假設如下(Yamada, Tokuno & Osaki, 1992)：

1. 軟體系統中的故障將會造成軟體隨機出現失效。
2. 每當發生軟體失效時，引起失效的軟體故障將被移除，但移除故障的過程中可能會產生新的故障。
3. 每單位測試時間偵測到的失效數量與軟體系統中目前的故障內容成正比。

以廣義的 NHPP 軟體可靠度模型表示，即假設 $a(t) = ae^{\alpha t}$ 、 $b(t) = b$ 以及 NHPP 基本的假設，初始值 $m(0) = 0$ 等條件，藉由求解微分方程(2.6)可得到：

$$m(t) = \frac{ab}{b+\alpha} (e^{\alpha t} - e^{-bt}) \quad (2.11)$$

六、H/D G-O 模型

H/D G-O 模型是由 Hossain 與 Dahiya 兩位學者於 1993 年所提出，該模型的假設如下(Hossain & Dahiya, 1993)：

1. 遵循 NHPP 過程。

2. 每當發生失效時，引起失效的故障將被程式設計師立即且完美地移除 (不會產生新的錯誤)。
3. 剩餘失效的偵測率是常數。

均值函數為：

$$m(t) = \ln \left(\frac{e^a - c}{e^{ae^{-bt}} - c} \right) \quad (2.12)$$

七、Pham-Zhang 模型

Pham-Zhang 模型，本文將此模型簡稱為 P-Z Model，P-Z Model 是由 Pham 與 Zhang 兩位學者於 1997 年所提出，經由 Pham 的整理，該模型的假設如下 (Pham, 2006)：

1. 錯誤偵測率會受故障影響而產生改變。
2. 每當軟體發生失效時，引起失效的軟體故障將立即被移除，並且可能產生新的故障。
3. 故障產生率為與測試時間相關的指數函數。換句話說，在測試過程開始時的故障產生速度大於結束時的故障產生速度。這反映了一個事實，即一開始在軟體中將產生更多的故障；而在最後，測試人員將擁有更多的知識，因此，程序中產生的故障將變得更少。
4. 錯誤偵測率函數為非遞減且帶有彎曲的 S 型模型。

以廣義的 NHPP 軟體可靠度模型表示，即假設 $a(t) = c + a(1 - e^{-at})$ 、 $b(t) = \frac{b}{1 + \beta e^{-bt}}$ 以及 NHPP 基本的假設，初始值 $m(0) = 0$ 等條件，藉由求解微分方程(2.6)可得到：

$$m(t) = \frac{(a+c)(1-e^{-bt}) - \frac{ab}{b-a}(e^{-at} - e^{-bt})}{1 + \beta e^{-bt}} \quad (2.13)$$

八、Pham-Nordmann-Zhang 模型

Pham-Nordmann-Zhang 模型，本文將此模型簡稱為 P-N-Z Model，P-N-Z

Model 是由 Pham、Nordmann 與 Zhang 三位學者於 1999 年所提出，該模型的假設如下(Pham, Nordmann & Zhang, 1999)：

1. 錯誤偵測率與時間有關。
2. 整體而言，故障內容函數與時間呈線性關係。
3. 在除錯過程中可能產生故障，故障的產生率為一常數。
4. 錯誤偵測率函數為非遞減的 S 型曲線，可藉由觀察錯誤偵測率函數截取到軟體測試人員的學習過程。

以廣義的 NHPP 軟體可靠度模型表示，即假設 $a(t) = a(1 + \alpha t)$ 、 $b(t) = \frac{b}{1 + \beta e^{-bt}}$ 以及 NHPP 基本的假設，初始值 $m(0) = 0$ 等條件，藉由求解微分方程(2.6)可得到：

$$m(t) = \frac{a\left(1 - \frac{\alpha}{b}\right)(1 - e^{-bt}) + a\alpha t}{1 + \beta e^{-bt}} \quad (2.14)$$

九、Pham Exponential Imperfect Debugging 模型

Pham Exponential Imperfect Debugging 模型，本文將此模型簡稱為 P-EID Model，P-EID Model 是由 Pham 於 2000 年所提出，經由 Pham 的整理，該模型的假設如下 (Pham, 2006)：

1. 錯誤偵測率會受故障影響而產生改變。
2. 每當軟體發生失效時，引起失效的軟體故障將立即被移除，並且可能產生新的故障。
3. 故障產生率是隨著測試時間而改變的指數函數。
4. 錯誤偵測率函數為非遞減的且帶有彎曲的 S 型模型。

以廣義的 NHPP 軟體可靠度模型表示，即假設 $a(t) = \alpha e^{\beta t}$ 、 $b(t) = \frac{b}{1 + ce^{-bt}}$ 以及 NHPP 基本的假設，初始值 $m(0) = 0$ 等條件，藉由求解微分方程(2.6)可得到：

$$m(t) = \frac{ab}{b+\beta} \frac{(e^{\beta t} - e^{-bt})}{1 + ce^{-bt}} \quad (2.15)$$

十、Roy 模型

Roy 模型是由 Roy、Mahapatra 與 Dey 三位學者於 2014 年所提出，該模型的假設如下(Roy, Mahapatra & Dey, 2014)：

1. 失效的觀察與排除現象遵循 NHPP。
2. 在軟體運行的過程中可能產生失效，而這些失效是由軟體中剩餘的故障所造成的。
3. 在測試過程中，錯誤偵測率是常數。
4. 軟體的失效率與軟體中剩餘的故障成正比。
5. 當故障被移除時，軟體中有機會產生新的故障。
6. 故障可能在測試的過程中產生，且除錯的過程中可能無法完全移除故障，即不完美除錯。
7. 在測試的一開始，測試團隊不太熟悉軟體系統，且他們對於如何正確除錯幾乎一無所知。隨著測試時間的增加，他們將獲得愈來愈多關於除錯的知識。
8. 故障內容函數是與測試時間相關的指數遞增函數。
9. 在測試開始時的故障產生率高於測試結束時的故障產生率。

以廣義的 NHPP 軟體可靠度模型表示，即假設 $a(t) = a(\alpha - e^{-\beta t})$ 、 $b(t) = b$ 以及 NHPP 基本的假設，初始值 $m(0) = 0$ 等條件，藉由求解微分方程(2.6)可得到：

$$m(t) = a\alpha(1 - e^{-bt}) - \frac{ab}{b-\beta}(e^{-\beta t} - e^{-bt}) \quad (2.16)$$

在軟體可靠度的領域中已有許多 NHPP 軟體可靠度成長模型被提出，而本研究僅舉例出十個模型做為參考，並於模型建構完畢後做為比較的模式使用。

第三章 研究方法

第一節 研究架構

本研究採用 NHPP 概念為基礎，藉由文獻探討汲取想法，經過發想之後，以導入函數概念的方式提出新的故障內容函數用於模型建構；模型建構完成之後，進行模型測試並且修正模型，利用參數估計方法估算出模型中各個參數，用以評估模型與失效數據之間的配適度。取得配適度資訊之後，再與現有的可靠度模型進行比較，以觀察新模型是否能更有效的預測軟體失效行為。

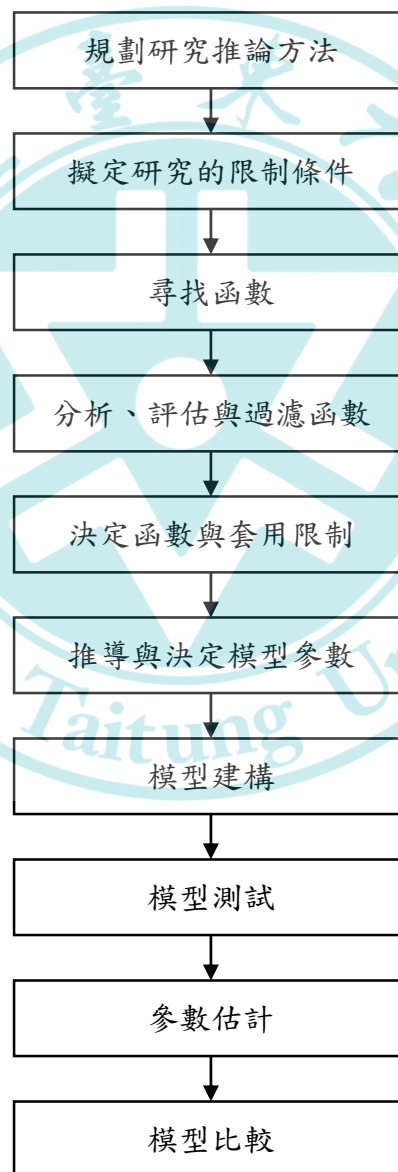


圖 9 研究架構圖

第二節 研究假設

表 2 研究假設中的變數說明

$m(t)$	基本公式的均值函數，定義為時間 t 時，期望偵測到的錯誤總數
$a(t)$	基本公式的故障內容函數，定義為時間 t 時，軟體累積的總錯誤數
$b(t)$	基本公式的錯誤偵測率函數，定義為時間 t 時，軟體的錯誤偵測率
$f(t)$	求解基本公式的均值函數時，所使用的積分因子函數
M1	本節推論過程中所假設的模型代稱
M2	本節推論過程中所假設的模型代稱
M3	本節推論過程中所假設的模型代稱，為模型 M1 與 M2 的組合
$m_1(t)$	模型 M1 的均值函數
$m_2(t)$	模型 M2 的均值函數
$m_3(t)$	模型 M3 的均值函數
$a_1(t)$	模型 M1 的故障內容函數
$a_2(t)$	模型 M2 的故障內容函數
$a_3(t)$	模型 M3 的故障內容函數
$b_1(t)$	模型 M1 的錯誤偵測率函數
$b_2(t)$	模型 M2 的錯誤偵測率函數
$b_3(t)$	模型 M3 的錯誤偵測率函數

資料來源：本研究整理

首先，從以往文獻中可發現許多軟體可靠度模型具有一種或多種函數特性，如：「具有除錯延遲與測試心力函數之不完美除錯軟體可靠度模型(李嘉華，2014)」，「具指數型、線性與 S 型特性之不完美除錯軟體可靠度模型(王亭雅，2015)」或是 P-Z Model(Pham & Zhang, 1997) …等等；因此本研究認為善用某些函數特性能夠有效的提升軟體可靠度模型的準確率。

然後，本研究觀察基本公式(2.6)：

$$\frac{dm(t)}{dt} = m'(t) = b(t)[a(t) - m(t)] \quad (3.1)$$

通過展開公式(3.1)，並移項可得到：

$$m'(t) + b(t)m(t) = a(t)b(t) \quad (3.2)$$

假設 $f(t) = e^{\int_{t_0}^t b(\tau)d\tau}$ ，並於公式(3.2)兩邊同乘上 $f(t)$ 可得：

$$m'(t)f(t) + b(t)f(t)m(t) = (f(t)m(t))' = a(t)b(t)f(t) \quad (3.3)$$

將公式(3.3)兩邊同時對 t 進行積分：

$$\int_{t_0}^t (f(\tau)m(\tau))' d\tau = f(t)m(t) - f(t_0)m(t_0) = \int_{t_0}^t a(\tau)b(\tau)f(\tau) d\tau \quad (3.4)$$

其中 $f(t_0) = 1$ 如公式(3.5)所示：

$$f(t_0) = e^{\int_{t_0}^{t_0} b(\tau)d\tau} = e^0 = 1 \quad (3.5)$$

通過移項與整理公式(3.4)可得：

$$m(t) = \frac{\int_{t_0}^t a(\tau)b(\tau)f(\tau)d\tau + m(t_0)}{f(t)} \quad (3.6)$$

獲得均值函數 $m(t)$ 後，本研究進行推論，令模型 M1 的故障內容函數為 $a_1(t)$ ，錯誤偵測率函數為 $b_1(t)$ ，均值函數為 $m_1(t)$ ，及初始條件 $m_1(0) = 0$ ；模型 M2 的故障內容函數為 $a_2(t)$ ，錯誤偵測率函數為 $b_2(t)$ ，均值函數為 $m_2(t)$ ，及初始條件 $m_2(0) = 0$ 。

假設 $\begin{cases} a_1(t) \neq a_2(t) \\ b_1(t) = b_2(t) = b_3(t) \end{cases}$ ， $a_1(t) \neq 0, a_2(t) \neq 0, b_3(t) \neq 0$ ；以及假設模型 M3 的故障內容函數為 $a_1(t) + a_2(t)$ ，錯誤偵測率函數為 $b_3(t)$ ，均值函數為 $m_3(t)$ ，及初始條件 $m_3(0) = 0$ ；最後假設 M1、M2 及 M3 三個模型運行於時間 $[0, t]$ 之間，且 $t \in \mathbb{N}$ 。根據公式(3.6)的推算可得以下結果：

$$m_1(t) = \frac{\int_0^t a_1(\tau)b(\tau)f(\tau)d\tau}{f(t)} \quad (3.7)$$

$$m_2(t) = \frac{\int_0^t a_2(\tau)b(\tau)f(\tau)d\tau}{f(t)} \quad (3.8)$$

$$\begin{aligned} m_3(t) &= \frac{\int_0^t (a_1(\tau) + a_2(\tau))b(\tau)f(\tau)d\tau}{f(t)} \\ &= \frac{\int_0^t a_1(\tau)b(\tau)f(\tau)d\tau}{f(t)} + \frac{\int_0^t a_2(\tau)b(\tau)f(\tau)d\tau}{f(t)} \\ &= m_1(t) + m_2(t) \end{aligned} \quad (3.9)$$

本研究透過推導及觀察基本公式發現，當兩個模型同樣運行於時間 $[0, t]$ 之間 $t \in \mathbb{N}$ ，且其錯誤偵測率函數相等時，兩故障內容函數相加後，所產生出新的

均值函數恰好等於兩個原均值函數的相加；此特性非常有利於導入新的函數，並使得模型能夠同時擁有兩種或是多種函數特性，因此本研究利用此特性，於故障內容函數中導入新的函數來建立新的軟體可靠度成長模型。

除此特性之外，從某些文獻中可發現學者對於故障內容函數的假設與失效強度函數或是均值函數直接相關；早期的軟體可靠度成長模型中，如：Y-LID Model (Yamada, Tokuno & Osaki, 1992) 或 P-Z Model (Pham & Zhang, 1997) 等，其故障內容函數的成長趨勢只受時間影響；而 Zhang、Teng 與 Pham 於 2003 年所提出的模型中，對於故障內容函數的假設為 $\frac{da(t)}{dt} = \beta(t) \frac{dm(t)}{dt}$ (Zhang, Teng & Pham, 2003)，此假設使得故障內容函數除了受時間影響之外，同時也受到失效強度函數的影響；在 2011 年，Kapur 與其他幾位學者提出的模型則假設故障內容函數 $A(t) = A + \alpha m(t)$ (Kapur et al., 2011a)，此假設使得故障內容函數受到時間以及均值函數的影響。

本研究由「故障內容函數受均值函數影響」的觀點出發，認為故障內容函數也受錯誤偵測率函數影響；基於基本假設，所有新產生的錯誤都是由軟體中剩餘的錯誤產生而來，而我們期望錯誤的偵測與清除將降低軟體中剩餘的錯誤數目，因此本研究假設錯誤偵測率函數對於故障內容函數的成長趨勢具有抑制效果。

第三節 函數導入

表 3 函數導入中的變數說明

$g(t)$	將導入故障內容函數的趨勢函數
$a(t)$	本研究規劃出的故障內容函數雛形
$b(t)$	本節推論過程中所假設的錯誤偵測率函數代稱
a	初始錯誤數
c	倍率調整函數

資料來源：本研究整理

基於上一節所述的特性及假設，配合微分方程的運算，在錯誤偵測率函數為非遞減函數的前提下，以錯誤偵測率函數對於故障內容函數具有抑制效果的觀點出發，本研究假設導入故障內容函數的函數基本形式為：

$$g(t) + \frac{g'(t)}{b(t)} \quad (3.10)$$

然後本研究參考 Kapur 等人所提出的模型(Kapur et al., 2011a)及 P-Z Model(Pham & Zhang, 1997)兩個模型對於故障內容函數的假設，將本篇論文所提模型的故障內容函數離形設計為：

$$a(t) = a + c \left(g(t) + \frac{g'(t)}{b(t)} - g(0) - \frac{g'(0)}{b(0)} \right) \quad (3.11)$$

限制條件為：

$$\begin{cases} 0 < a \\ 0 < b(t) \leq 1, & \forall t \in \mathbb{N} \\ \exists g(t): 0 \leq \frac{da(t)}{dt}, & \forall t \in \mathbb{N} \end{cases} \quad (3.12)$$

此故障內容函數的設計能夠使得初始錯誤數目 $a(0)$ 恰好等於 a ，有利於表達初始錯誤的數量；而限制條件的解釋如以下三點：

1. 初始錯誤數目 a 需大於 0：基於基本假設，所有新產生的錯誤都是由軟體中剩餘的錯誤產生而來，若初始錯誤數目為 0 個錯誤，則往後也將不再產生新的錯誤，此情況下不再需要除錯；初始錯誤數目少於 0 個是不合常理的情況，雖然從數學的角度，此模型仍可運算，但因與實際情況背離，因此不被接受。
2. 錯誤偵測率函數任何時候都需大於 0：限縮錯誤偵測率函數主要是為配合假設，在本研究的假設中，若錯誤偵測率函數為 0，將造成整個故障內容函數失去意義；不過就錯誤偵測率函數本身而言，單一時間點為 0 是被允許的，此僅代表於該時間點沒有發現任何錯誤，但錯誤偵測率函數為零函數是不被接受的，這代表任何錯誤都不會被發現，表示除錯永不可能完成。
3. 故障內容函數的微分需大於或等於 0：由於故障內容函數是錯誤累積函數，它是一個隨著時間增加而增加的遞增函數，表示其在任一時間點的變化量都將大於或等於 0，因此利用微分表示故障內容函數的瞬時變化量，並限制其必需大於或等於 0，以確保故障內容函數的遞增趨勢。

第四節 賦予學習效果

表 4 賦予學習效果中的變數說明

M4	本節推論過程中所假設的模型代稱
M_L	本節推論過程中所假設的模型代稱，由 M4 模型進一步推算得出
$M_{initial}$	本節推論過程中所假設的模型代稱，為未具備學習效果的初始模型
M	本節推論過程中所假設的模型代稱，為被賦予學習效果後的新模型
$m_4(t)$	模型 M4 的均值函數
$m_L(t)$	模型 M_L 的均值函數
$m_{initial}(t)$	模型 $M_{initial}$ 的均值函數
$m(t)$	模型 M 的均值函數
$a_4(t)$	模型 M4 的障內容函數
$a_L(t)$	模型 M_L 的障內容函數
$a_{initial}(t)$	模型 $M_{initial}$ 的障內容函數
$a(t)$	模型 M 的障內容函數
$b_4(t)$	模型 M4 的錯誤偵測率函數
$b_L(t)$	模型 M_L 的錯誤偵測率函數
$b_{initial}(t)$	模型 $M_{initial}$ 的錯誤偵測率函數
$b(t)$	模型 M 的錯誤偵測率函數
$L(t)$	學習效果的賦予函數
$z(t)$	求解微分方程 $L(t)$ 時，所使用的積分因子函數
c	求解微分方程 $L(t)$ 時，所使用的積分常數
$X(t)$	嘗試解析學習效果的學習曲線模擬函數
x_1	函數 $X(t)$ 之變數，解釋為學習的極限
x_2	函數 $X(t)$ 之變數，解釋為學習的效率
x_3	函數 $X(t)$ 之變數，解釋為學習的經驗

資料來源：本研究整理

本研究在觀察現有模型的時候，發現了某些模型之間具有相關聯性，關於該特性在條件限制的情況下，推論過程如下：

本研究令模型 M4 的故障內容函數為 $a_4(t)$ ，錯誤偵測率函數為 $b_4(t)$ ，均值函數為 $m_4(t)$ ，及初始條件 $m_4(0) = 0$ 。

假設存在函數 $L(t)$ ，使模型 M_L 成立；模型 M_L 的故障內容函數為 $a_L(t) = a_4(t)$ ，錯誤偵測率函數為 $b_L(t) = b_4(t) \times L(t)$ ，均值函數為 $m_L(t) = m_4(t) \times L(t)$ ，及初始條件 $m_L(0) = 0$ ；最後假設 M4 及 M_L 兩個模型運行於時間 $[0, t]$ 之間 $t \in \mathbb{N}$ 。

求取函數 $L(t)$ ，首先將模型 M4 帶入基本公式(3.1)可得：

$$m_4'(t) = b_4(t)[a_4(t) - m_4(t)] \quad (3.13)$$

將公式(3.13)兩邊同除掉 $b_4(t)$ 再加上 $m_4(t)$ 可得：

$$a_4(t) = \frac{m_4'(t)}{b_4(t)} + m_4(t) \quad (3.14)$$

再將模型 M_L 帶入基本公式(3.1)可得：

$$m_L'(t) = b_L(t)[a_L(t) - m_L(t)] \quad (3.15)$$

將假設條件帶入公式(3.15)，展開並整理，則可得到以下公式：

$$m_4'(t)L(t) + m_4(t)L'(t) = a_4(t)b_4(t)L(t) - b_4(t)m_4(t)L^2(t) \quad (3.16)$$

然後再將公式(3.14)帶入公式(3.16)並整理，即可得函數 $L(t)$ 的微分方程如下：

$$L'(t) = b_4(t)L(t) - b_4(t)L^2(t) = b_4(t)L(t)(1 - L(t)) \quad (3.17)$$

解函數 $L(t)$ 的微分方程，首先令 $L(t) = ce^{\int_0^t z(\tau)d\tau}$, $c \in \mathbb{R}, c \neq 0$ 帶入(3.17)可得：

$$z(t)ce^{\int_0^t z(\tau)d\tau} = b_4(t)ce^{\int_0^t z(\tau)d\tau} \left(1 - ce^{\int_0^t z(\tau)d\tau}\right) \quad (3.18)$$

將公式(3.18)兩邊同除掉 $ce^{\int_0^t z(\tau)d\tau}$ 可得：

$$z(t) = b_4(t) \left(1 - ce^{\int_0^t z(\tau)d\tau}\right) \quad (3.19)$$

將公式(3.19)兩邊同除掉 $1 - ce^{\int_0^t z(\tau)d\tau}$ 可得：

$$b_4(t) = \frac{z(t)}{1 - ce^{\int_0^t z(\tau)d\tau}} = - \left(\frac{-z(t)e^{-\int_0^t z(\tau)d\tau}}{e^{-\int_0^t z(\tau)d\tau} - c} \right) \quad (3.20)$$

將公式(3.20)兩邊同時對 t 進行積分可得：

$$\int_0^t b_4(\tau) d\tau = \int_0^t - \left(\frac{-z(\tau)e^{-\int_0^\tau z(\bar{\tau})d\bar{\tau}}}{e^{-\int_0^\tau z(\bar{\tau})d\bar{\tau}} - c} \right) d\tau = - \left(\ln \left(e^{-\int_0^t z(\bar{\tau})d\bar{\tau}} - c \right) \right) \Big|_0^t \quad (3.21)$$

將公式(3.21)進行整理後可得：

$$\int_0^t b_4(\tau) d\tau = \ln\left(\frac{1-c}{e^{-\int_0^t z(\tau)d\tau}-c}\right) \quad (3.22)$$

將公式(3.22)兩邊取指數可得：

$$e^{\int_0^t b_4(\tau)d\tau} = e^{\ln\left(\frac{1-c}{e^{-\int_0^t z(\tau)d\tau}-c}\right)} = \frac{1-c}{e^{-\int_0^t z(\tau)d\tau}-c} \quad (3.23)$$

將公式(3.23)進行整理後可得：

$$e^{\int_0^t z(\tau)d\tau} = \frac{1}{(1-c)e^{-\int_0^t b_4(\tau)d\tau}+c} \quad (3.24)$$

將公式(3.24)兩邊同乘上 c 後可得：

$$ce^{\int_0^t z(\tau)d\tau} = \frac{c}{(1-c)e^{-\int_0^t b_4(\tau)d\tau}+c} = \frac{1}{1+\left(\frac{1-c}{c}\right)e^{-\int_0^t b_4(\tau)d\tau}} = L(t) \quad (3.25)$$

最後我們求得函數 $L(t)$ 如下：

$$L(t) = \frac{1}{1+\left(\frac{1-c}{c}\right)e^{-\int_0^t b_4(\tau)d\tau}} \quad (3.26)$$

因此，我們得到 M_L 模型，其故障內容函數、錯誤偵測率以及均值函數如下：

$$a_L(t) = a_4(t) \quad (3.27)$$

$$b_L(t) = b_4(t) \times L(t) = \frac{b_4(t)}{1+\left(\frac{1-c}{c}\right)e^{-\int_0^t b_4(\tau)d\tau}} \quad (3.28)$$

$$m_L(t) = m_4(t) \times L(t) = \frac{m_4(t)}{1+\left(\frac{1-c}{c}\right)e^{-\int_0^t b_4(\tau)d\tau}} \quad (3.29)$$

此時我們可以得知，可能存在某一函數 $L(t)$ 乘上錯誤偵測率函數後，能夠在不改變故障內容函數的情況下，恰好使新的均值函數等於原均值函數乘上 $L(t)$ 函數。

而關於學習效果，在各個領域中可能有各種不同的呈現方式；在此，本研究依據 P-N-Z Model 的假設進行分析，該模型對於錯誤偵測率函數的假設為 $b(t) = \frac{b}{1+\beta e^{-bt}}$ ，對於該函數的描述為：「錯誤偵測率函數是非遞減的 S 型曲線，可以記錄軟體測試人員的學習過程」(Pham, Nordmann & Zhang, 1999)，此過程表現了錯誤偵測率隨著測試時間增加而增加的現象，函數呈現 S 型遞增，正好呼應了文獻探討第七節學習效果中的正負加速變化，驗證了測試人員於偵錯的過程中具有學習的現象。但三位學者並未詳細說明函數中各個參數所代表的意義，因此，本研究從數學的角度出發，分析各個參數的限制，並嘗試對其進行解釋，過程如下：

首先，本研究根據 P-N-Z 模型的錯誤偵測率函數模擬了一個函數 $X(t)$ ，並假設它能夠描述學習效果，也就是隨時間增加而遞增的 S 型曲線，函數如下：

$$X(t) = \frac{x_1}{1+x_2 e^{-x_3 t}} \quad (3.30)$$

初始限制為 x_1 、 x_2 和 x_3 為實數， t 為包含零的正整數， t 於模型中代表時間，因此只考慮 $t \geq 0$ 的情況。

經由觀察分析可發現，若 $x_1 < 0$ 將導致整個 $X(t)$ 函數的數值皆為負值或是失去 S 型遞增的函數特性，若 $x_1 = 0$ ，則函數 $X(t)$ 為零函數；因此限制 $x_1 > 0$ ；若 $x_2 < 0$ 將導致函數 $X(t)$ 無法保持 S 型曲線的特性，若 $x_2 = 0$ ，則函數 $X(t)$ 為水平線；因此限制 $x_2 > 0$ 。若 $x_3 < 0$ 將導致函數 $X(t)$ 無法保持隨時間增加而遞增特性，若 $x_3 = 0$ ，則函數 $X(t)$ 為水平線；因此限制 $x_3 > 0$ 。

因此，限制條件修正限縮為 x_1 、 x_2 和 x_3 為正實數， t 為正整數。

進一步分析可發現：

$$\lim_{t \rightarrow \infty} X(t) = x_1 \quad (3.31)$$

x_1 為函數 $X(t)$ 的上限值；而 x_2 為 t 的係數，直接影響函數的走勢， x_2 的數值愈大，函數 $X(t)$ 的上升速度愈快，收斂速度也愈快； x_3 則具有平移的效果， x_3 的數值愈大，函數 $X(t)$ 向右偏移愈多。

而本研究嘗試以學習效果的角度針對函數 $X(t)$ 的各項參數進行解釋，解釋內容如下：

1. x_1 為函數 $X(t)$ 的上限值，本研究將其解釋為學習效果的極限值。
2. x_2 直接影響函數 $X(t)$ 的趨勢，本研究將其解釋為學習的效率，學習效率愈高，進步愈快，但同時也愈快達到極限。
3. x_3 為函數 $X(t)$ 的平移調整參數，因此本研究將其解釋為經驗值， x_3 的數值愈大，表示先前的學習經驗愈少，所需更長的學習歷程才可達到學習的極限；反之，若先前就有較多經驗， x_3 的數值愈接近 0，函數 $X(t)$ 向左移，所需的學習歷程愈短，可用較少的時間達到學習的極限。

※備註：當 x_3 很大或是趨近於 0 時，S 型函數的趨勢變化都相對的小，本研究認為此兩種情況都不利於觀察學習歷程。

以上述解釋來分析 P-N-Z Model，它的錯誤偵測率函數為 $b(t) = \frac{b}{1+\beta e^{-bt}}$ ；其中 b 同時代表學習的上限以及學習效率，加上錯誤偵測率的限制使得 $0 < b \leq 1$ ；而 β 代表經驗值， β 愈接近 0，表示經驗愈豐富， $\beta > 0$ 。

此時，我們觀察本節一開始的推論，假設 $b_4(t) = b$ ，帶入公式(3.28)可得：

$$b_L(t) = \frac{b_4(t)}{1 + \left(\frac{1-c}{c}\right)e^{-\int_0^t b_4(\tau) d\tau}} = \frac{b}{1 + \frac{(1-c)}{c}e^{-bt}} \quad (3.32)$$

當 $0 < c < 1$ 時， $\frac{(1-c)}{c}$ 恰好屬於正實數，與 β 的值域相同；因此，本研究推論，當滿足以上條件時， $b_L(t)$ 與 P-N-Z Model 的錯誤偵測率函數同樣具備描述學習效果的能力。

故本研究推論，在故障內容函數合理存在時，若能滿足以下兩個條件：

1. 初始的錯誤偵測率函數假設為 $b_{initial}(t) = b, b \in (0,1)$
2. 公式(3.26)中的 c 界於 0 到 1 之間，使得 $\frac{(1-c)}{c} = \beta > 0$

則學習效果將是可以被賦予的。賦予學習的過程如下：

假設模型 $M_{initial}$ 成立；模型 $M_{initial}$ 的故障內容函數為 $a_{initial}(t)$ ，錯誤偵測率函數為 $b_{initial}(t) = b$ ，均值函數為 $m_{initial}(t)$ ，以及初始條件 $m_{initial}(0) = 0$ ；並且存在模型 M ，其故障內容函數為 $a(t) = a_{initial}(t)$ 、錯誤偵測率函數為 $b(t)$ ，均值函數為 $m(t)$ ，以及初始條件 $m(0) = 0$ ；然後假設 $M_{initial}$ 模型與 M 模型皆運行於時間 $[0, t]$ 之間 $t \in \mathbb{N}$ 。

利用公式(3.28)指定 $b(t) = \frac{b_{initial}(t)}{1 + \left(\frac{1-c}{c}\right)e^{-\int_0^t b_{initial}(\tau) d\tau}}$ ，並假設 $0 < c < 1$ ，此時 $\frac{(1-c)}{c} = \beta$ ，使得 $b(t) = \frac{b}{1 + \beta e^{-bt}}$ ， $\beta > 0$ ；利用公式(3.29)及以上假設條件可得：

$$m(t) = \frac{m_{initial}(t)}{1 + \beta e^{-bt}} \quad (3.33)$$

藉由以上推論與整理，本研究將於模型建構時，利用此方法賦予模型學習效果，使模型更加符合現實情境，期望達到提升模型準確率的效果。

第五節 資料來源

本研究為驗證模型的準確性，透過七組實際的失效數據進行模型的參數估計，並與現有模型進行比較與評估。其中所使用的七組失效數據為已公開發表的論文中過去學者們所使用的失效數據集。所使用失效數據資料來源如表 2 所示。

表 5 失效數據來源

資料集	失效數據來源
一	Ohba 的測試資料(Ohba, 1984b)
二	即時指揮與控制系統(Musa, 1987)
三	Tandem Computers 的軟體產品(Wood, 1996)
四	無線網路交換中心(Jeske & Zhang, 2005)
五	即時控制的應用程式(Lyu, 1996)
六	即時監視與控制系統(Lyu, 1996)。
七	鐵路聯鎖(Interlocking)系統(Lyu, 1996)。

資料來源：本研究整理

第六節 參數估計方法

參數估計方法的使用目的是為了估算出能使模型較符合實際數據的參數，以下說明兩種方法，於模型建構完成之後，用來估算模型參數，分別是最小平方估計法以及最大似然估計法 (Maximum Likelihood Estimation, MLE)。

一、最小平方估計法

最小平方估計法，以線性問題為例；首先假設 X 和 $E(Y|x)$ 之間存在線性關係，即 $E(Y|x) = a + bx$ 。給定一組數據，估計參數 a 和 b 以求取最小化平方和。假設所需多項式 $p(x) = \sum_{i=0}^m a_i x^i$ ，其中 $a_0, a_1, \dots, a_m$ 有待確定。最小平方估計法將選擇能使數據點到 $p(x)$ 垂直距離的平方和最小化的係數為解。表示最佳的參數 a 和 b 能使 $L = \sum_{i=1}^n (y_i - p(x_i))^2$ 最小化。此處只考慮線性的情況，假設直線 $X = \alpha + \beta Z$ 對於每項的觀察 $(x_i, z_i): X_i = \alpha + \beta Z_i$ ，令 $Q = \sum_{i=1}^n (\alpha + \beta z_i)^2$ ，而我們希望找到 α 和 β 的估計值，能使 Q 最小化，因此我們獲得(Pham, 2006)：

$$\frac{\partial Q}{\partial \alpha} = -2 \sum_{i=1}^n (\alpha + \beta z_i) = 0 \quad (3.34)$$

$$\frac{\partial Q}{\partial \beta} = -2 \sum_{i=1}^n z_i (\alpha + \beta z_i) = 0 \quad (3.35)$$

二、最大似然估計法

提到最大似然估計法，首先必須先認識似然函數(Likelihood function)，其定義如下(Watkins, 2016)：

在古典統計中，自然狀態(State of Nature)被假設為固定的，但我們並不清楚；因此，估計的其中一個目的就是為了確認 P_θ 是資料的來源。所以，「估計」是一種統計，表示為 $\hat{\theta}: data \rightarrow \theta$ 。而對於每項參數數值 $\theta \in \theta$ 存在一個密度函數，表示為 $f_X(x|\theta)$ 。假設資料是源自於隨機變數序列的樣本 $X_1(\omega), \dots, X_n(\omega)$ ，是由具備共同密函數 $f_X(x|\theta)$ 的分佈家族中得出，其中參數值 θ 必須估算才可得知。由於其中各個隨機變數是獨立的，所以聯合密度(Joint Density)是邊際密度(Marginal Densities)的乘積，表示為：

$$f_X(x|\theta) = \prod_{k=1}^n f_X(x_k|\theta) = f_X(x_1|\theta) f_X(x_2|\theta) \dots f_X(x_n|\theta) \quad (3.36)$$

在此種情況下，資料 x 為已知，但參數 θ 是未知的。因此，密度函數被表示為 $L(\theta|x) = f_X(x|\theta)$ ，而其中 L 被稱為似然函數。

而最大似然估計法是藉由觀察一系列概率 P_θ 中選擇的隨機變數 $X =$

$(X_1, X_2, \dots, X_n)$ 為起始，當 θ 為自然狀態時， $f(x|\theta)$ 和 $x = (x_1, x_2, \dots, x_n)$ 被用表示資料的密度函數。然後，藉由最大似然原理可得到 $\hat{\theta}$ 做為觀測資料最有可能出現的參數估計值。其中，似然函數被視為 θ 的密度函數，表示為(Watkins, 2016)：

$$L(\theta|x) = f(x|\theta), \theta \in \Theta \quad (3.37)$$

而最大似然估計表示為：

$$\hat{\theta}(x) = \arg \max_{\theta} L(\theta|x) \quad (3.38)$$

由於最小平方估計法能夠容易求出各模型之未知參數的估計值，並使其與實際數據之間的誤差平方和最小，因此本研究主要利用最小平方估計法求取各模型之參數，並繪製出各模型之配適度曲線圖。

第七節 模型評估標準

為了評估本研究所提出的模型是否能夠有效的預測軟體的失效情況，本研究採用五項評估標準來分析本研究所提出的模型預測數值與實際失效數據之間的差異，並與文獻探討第八節中所提到的十個現有的模型進行比較。五項評估標準分別為均方誤差(Mean Square Error, MSE)、多元判定係數 (Coefficient of Multiple Determination, R^2)、絕對誤差之和(Sum of Absolute Error, SAE)、預測相對變量(Predicted Relative Variation, PRV)和均方根預測誤差 (Root Mean Square Prediction Error, RMSPE)，詳細描述如下：

表 6 模型評估標準中的變數說明

n	觀測數據點的總數。
t_i	第 i 個時間點。 $(i = 0, 1, 2, 3, \dots, n)$
$m(t_i)$	在第 i 個時間點，模型所預測的累積失效數目。
$y(t_i)$	在第 i 個時間點，實際觀測到的累積失效數目。
k	模型中所使用的參數個數。
$\bar{y}$	實際觀測的累積失效數據的平均數。
$PE(t_i)$	在第 i 個時間點的預測誤差(Prediction Errors)，
Bias	預測誤差的平均數。

資料來源：本研究整理

一、均方誤差

均方誤差 (MSE) 可測量模型的預測值與實際觀測值之間的偏差以及參數數量的影響，Pham 於 2016 年將此方法做為軟體可靠度成長模型的模型評估標準，其定義如下(Pham, 2016)：

$$MSE = \frac{\sum_{i=1}^n (y(t_i) - m(t_i))^2}{n-k} \quad (3.39)$$

MSE 值愈小，表示模型的配適度誤差愈低，代表模型的配適度愈好。

二、多元判定係數

多元判定係數(R^2)可用來衡量模型對於真實數據變化的匹配成功程度，Park 與 Baik 於 2015 年將此方法做為軟體可靠度成長模型的模型評估標準，其定義如下(Park & Baik, 2015)：

$$\bar{y} = \frac{\sum_{i=1}^n y(t_i)}{n} \quad (3.40)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y(t_i) - m(t_i))^2}{\sum_{i=1}^n (y(t_i) - \bar{y})^2} \quad (3.41)$$

R^2 的數值介於 0 到 1 之間，愈接近 1 表示模型的預測與實際數據的匹配成功程度愈高，代表模型的配適度愈好。

三、絕對誤差之和

絕對誤差之和(SAE)，它是用來測量模型所預測的失效數量與實際觀測數據之間的差距，Lee、Chang、Pham 與 Song 於 2018 年將此方法做為軟體可靠度成長模型的模型評估標準，其定義如下(Lee, Chang, Pham & Song, 2018)：

$$SAE = \sum_{i=1}^n |y(t_i) - m(t_i)| \quad (3.42)$$

SAE 值愈小，表示模型的預測值與實際數據愈接近，代表模型的配適度愈好。

四、預測相對變量

預測相對變量(PRV)是預測誤差的標準差，Xu 與 Yao 於 2016 年將此方法做為軟體可靠度成長模型的模型評估標準，其定義如下(Xu & Yao, 2016)：

$$PE(t_i) = y(t_i) - m(t_i) \quad (3.43)$$

$$Bias = \frac{\sum_{i=1}^n PE(t_i)}{n} \quad (3.44)$$

$$PRV = \sqrt{\frac{\sum_{i=1}^n (PE(t_i) - Bias)^2}{n-1}} \quad (3.45)$$

PRV 值愈低，表示模型預測誤差的標準差愈小，代表模型的配適度愈好。

五、均方根預測誤差

均方根預測誤差(RMSPE)是用來測量模型所預測的失效數量與實際觀測數據之間的接近程度，可利用 Bias 和 PRV 求得，Xu 與 Yao 於 2016 年將此方法做為軟體可靠度成長模型的模型評估標準，其定義如下(Xu & Yao, 2016)：

$$RMSPE = \sqrt{Bias^2 + PRV^2} \quad (3.46)$$

RMSPE 值愈低，表示模型的預測與實際觀測數據的接近程度愈高，代表模型的配適度愈好。

第八節 模型建構

表 7 模型建構中的變數說明

$m(t)$	均值函數：時間 t 時，期望偵測到的錯誤總數
$m_{initial}(t)$	未具備學習效果的均值函數
$a(t)$	故障內容函數：時間 t 時，軟體累積的總錯誤數
a	軟體的初始錯誤數($a > 0$)
$b(t)$	錯誤偵測率函數：時間 t 時，軟體的錯誤偵測率
$b_{initial}(t)$	未賦予學習效果的錯誤偵測率函數
b	錯誤偵測率($0 < b \leq 1$)
c	倍率調整常數($c > 0$)
α	錯誤產生率($0 < \alpha \leq 2b$)(限制條件制定請參見附錄 1)
β	曲折因子($\beta > 0$)
γ	平移調整常數

資料來源：本研究整理

一、問題簡述

建立軟體可靠度模型期望的是能有效率的預估軟體失效的發生，而在模型之中，函數的特性對於模型的預測能力有巨大的影響，從文獻探究的過程中可發現，NHPPSRGMs 的發展脈絡，由一開始的指數型態，發展為 S 型型態，到後來的多種複合型態和更複雜的函數型態等，此現象啟發了本研究提出導入函數的想法。

首先，本研究以三角函數(Trigonometric Function)、雙曲函數(Hyperbolic Function)、反三角函數(Inverse Trigonometric Function)與反雙曲函數(Inverse Hyperbolic Function)為目標進行測試，過濾掉趨勢與限制條件不符合的函數後，最終由剩餘的函數中挑選了反雙曲正弦函數(Inverse Hyperbolic Sine Function, arsinh)做為導入模型的重要函數；利用第三章所提之方法將模型建構完畢後，藉由實際失效資料的測試，進一步分析以及與現有模型比較，判斷本研究所提模型(Arsinh Model)是否能夠獲得更好的配適度。

二、模型基本假設

1. 錯誤偵測的過程符合 NHPP 之形式。
2. 軟體運行過程中，任何時間下所產生的失效都是由未被偵測到的剩餘故障所產生。
3. 故障內容函數為遞增函數，將隨著時間增加而增加。
4. 故障內容函數的初始值為一隨機變數 a ($a > 0$)。
5. 在軟體除錯的過程當中，偵測到的錯誤將會被立刻清除，且仍可能會產生新的錯誤，即不完美除錯。
6. 錯誤偵測率函數是一個非遞減的 S 型曲線函數，當時間增加，錯誤偵測率會隨之漸增，可取得軟體測試者的學習過程。
7. 在測試的過程中，測試團隊起初對軟體並不熟悉，他們並不知道如何正確除錯，也沒有如何正確除錯的相關知識，但隨著測試時間的增加，他們將獲得了越來越多關於正確除錯的知識。

三、模型建構

首先，令初始錯誤偵測率函數 $b_{initial}(t) = b$ ，並根據公式(3.11)的假設，帶進所要導入的函數 $g(t) = \text{arsinh}(\alpha t + \gamma)$ 可得到：

$$a(t) = a + c \left[\text{arsinh}(\alpha t + \gamma) + \frac{\alpha}{b\sqrt{(\alpha t + \gamma)^2 + 1}} - \text{arsinh}(\gamma) - \frac{\alpha}{b\sqrt{\gamma^2 + 1}} \right] \quad (3.47)$$

藉由 $a(t)$ 、 $b_{initial}(t)$ 以及初始值 $m_{initial}(0) = 0$ 等條件，透過解微分方程(2.6)可得到 $m_{initial}(t)$ 如下：

$$m_{initial}(t) = \left(a - c \frac{\alpha}{b\sqrt{\gamma^2 + 1}} \right) (1 - e^{-bt}) + c(\text{arsinh}(\alpha t + \gamma) - \text{arsinh}(\gamma)) \quad (3.48)$$

然後透過公式(3.27)配合賦予學習效果所需的限制條件可得到錯誤偵測率函數 $b(t)$ 如下：

$$b(t) = \frac{b_{initial}(t)}{1 + \left(\frac{1-c}{c} \right) e^{-\int_0^t b_{initial}(\tau) d\tau}} = \frac{b}{1 + \beta e^{-bt}}, \text{ with } 0 < c < 1, \beta > 0 \quad (3.49)$$

因此，藉由以上假設與方法建立的故障內容函數及錯誤偵測率函數如下：

$$a(t) = a + c \left[\text{arsinh}(\alpha t + \gamma) + \frac{\alpha}{b\sqrt{(\alpha t + \gamma)^2 + 1}} - \text{arsinh}(\gamma) - \frac{\alpha}{b\sqrt{\gamma^2 + 1}} \right] \quad (3.50)$$

和

$$b(t) = \frac{b}{1 + \beta e^{-bt}} \quad (3.51)$$

此時將 $a(t)$ 和 $b(t)$ 以及初始值 $m(0) = 0$ 等條件帶入公式(2.6)求解微分方程；亦或是利用公式(3.33) 賦予學習效果的模式配合使用公式(4.2)賦予學習效果，兩方法皆可獲得 Arsinh 模型的均值函數，其均值函數 $m(t)$ 如下：

$$m(t) = \frac{\left(a - c \frac{\alpha}{b\sqrt{\gamma^2 + 1}} \right) (1 - e^{-bt}) + c(\text{arsinh}(\alpha t + \gamma) - \text{arsinh}(\gamma))}{1 + \beta e^{-bt}} \quad (3.52)$$

第四章 模型測試與資料分析

為了評估建立模型的有效性，本研究運用七組實際失效數據，以最小平方估計法求取模型的各個參數，再以 MSE、 R^2 、SAE、PRV 及 RMSPE 五個標準來評估模型的配適度。而其他用於比較的現有模型，其參數估計方法同於所提模型，因此本研究僅列出所提模型的參數估計過程。

第一節 參數估計

假設 $a \cdot b \cdot c \cdot \alpha \cdot \beta \cdot \gamma$ 是由 n 組數據： $\{(t_0, m_0), (t_1, m_1), (t_2, m_2), \dots, (t_n, m_n)\}$ 所決定，其中 $m_i (i \in \{0, 1, 2, \dots, n\})$ 為時間 $(0, t_i)$ 內偵測到的所有錯誤數目，透過最小平方估計法，可獲得模型估計函數(Evaluation Function)如下：

$$\begin{aligned} & \text{Min } M(a \cdot b \cdot c \cdot \alpha \cdot \beta \cdot \gamma) \\ &= \sum_{i=1}^n (m_i - m(t_i))^2 \\ &= \sum_{i=1}^n \left(m_i - \frac{\left(\left(a - c \frac{\alpha}{b \sqrt{(\gamma)^2 + 1}} \right) (1 - e^{-bt_i}) + c(\text{arsinh}(\alpha t_i + \gamma) - \text{arsinh}(\gamma)) \right)}{1 + \beta e^{-bt_i}} \right)^2 \end{aligned} \quad (4.1)$$

透過對方程式中的各個參數進行微分，並令其偏導數(Partial Derivatives)為零，即可獲得下列方程式：

$$\begin{aligned} & \frac{\partial M(a \cdot b \cdot c \cdot \alpha \cdot \beta \cdot \gamma)}{\partial a} \\ &= -2 \sum_{i=1}^n \left(\left(m_i - \frac{\left(\left(a - c \frac{\alpha}{b \sqrt{(\gamma)^2 + 1}} \right) (1 - e^{-bt_i}) + c(\text{arsinh}(\alpha t_i + \gamma) - \text{arsinh}(\gamma)) \right)}{1 + \beta e^{-bt_i}} \right) \times \left(\frac{1 - e^{-bt_i}}{1 + \beta e^{-bt_i}} \right) \right) = 0 \end{aligned} \quad (4.2)$$

$$\begin{aligned}
& \frac{\partial M(a \cdot b \cdot c \cdot \alpha \cdot \beta \cdot \gamma)}{\partial b} \\
&= -2 \sum_{i=1}^n \left(\left(m_i - \frac{\left(a - c \frac{\alpha}{b \sqrt{(\gamma)^2 + 1}} \right) (1 - e^{-bt_i}) + c(\operatorname{arsinh}(\alpha t_i + \gamma) - \operatorname{arsinh}(\gamma))}{1 + \beta e^{-bt_i}} \right) \right. \\
& \quad \times \left(\frac{t_i \left(a - \frac{c\alpha}{b \sqrt{(\gamma)^2 + 1}} \right) e^{-bt_i} + \frac{c\alpha(1 - e^{-bt_i})}{b^2 \sqrt{(\gamma)^2 + 1}}}{1 + \beta e^{-bt_i}} \right. \\
& \quad \left. \left. + \frac{\beta t_i \left(\left(a - \frac{c\alpha}{b \sqrt{(\gamma)^2 + 1}} \right) (1 - e^{-bt_i}) + c(\operatorname{arsinh}(\alpha t_i + \gamma) - \operatorname{arsinh}(\gamma)) \right)}{e^{bt_i} (1 + \beta e^{-bt_i})^2} \right) \right) = 0 \quad (4.3)
\end{aligned}$$

$$\begin{aligned}
& \frac{\partial M(a \cdot b \cdot c \cdot \alpha \cdot \beta \cdot \gamma)}{\partial c} \\
&= -2 \sum_{i=1}^n \left(\left(m_i - \frac{\left(a - c \frac{\alpha}{b \sqrt{(\gamma)^2 + 1}} \right) (1 - e^{-bt_i}) + c(\operatorname{arsinh}(\alpha t_i + \gamma) - \operatorname{arsinh}(\gamma))}{1 + \beta e^{-bt_i}} \right) \right. \\
& \quad \times \left(\frac{\operatorname{arsinh}(\alpha t_i + \gamma) - \operatorname{arsinh}(\gamma) + \frac{\alpha(e^{-bt_i} - 1)}{b \sqrt{(\gamma)^2 + 1}}}{1 + \beta e^{-bt_i}} \right) \left. \right) = 0 \quad (4.4)
\end{aligned}$$

$$\begin{aligned}
& \frac{\partial M(a \cdot b \cdot c \cdot \alpha \cdot \beta \cdot \gamma)}{\partial \alpha} \\
&= -2 \sum_{i=1}^n \left(\left(m_i - \frac{\left(a - c \frac{\alpha}{b \sqrt{(\gamma)^2 + 1}} \right) (1 - e^{-bt_i}) + c(\operatorname{arsinh}(\alpha t_i + \gamma) - \operatorname{arsinh}(\gamma))}{1 + \beta e^{-bt_i}} \right) \right. \\
& \quad \times \left(\frac{\frac{ct_i}{b \sqrt{(\alpha t_i + \gamma)^2 + 1}} + \frac{c(e^{-bt_i} - 1)}{b \sqrt{(\gamma)^2 + 1}}}{1 + \beta e^{-bt_i}} \right) \left. \right) = 0 \quad (4.5)
\end{aligned}$$

$$\begin{aligned}
& \frac{\partial M(a \cdot b \cdot c \cdot \alpha \cdot \beta \cdot \gamma)}{\partial \beta} \\
&= -2 \sum_{i=1}^n \left(\left(m_i - \frac{\left(a - c \frac{\alpha}{b \sqrt{(\gamma)^2 + 1}} \right) (1 - e^{-bt_i}) + c(\operatorname{arsinh}(\alpha t_i + \gamma) - \operatorname{arsinh}(\gamma))}{1 + \beta e^{-bt_i}} \right) \right. \\
& \quad \times \left. \frac{\left(a - \frac{c\alpha}{b \sqrt{(\gamma)^2 + 1}} \right) (e^{-bt_i} - 1) - c(\operatorname{arsinh}(\alpha t_i + \gamma) - \operatorname{arsinh}(\gamma))}{e^{bt_i} (1 + \beta e^{-bt_i})^2} \right) = 0 \quad (4.6)
\end{aligned}$$

$$\begin{aligned}
& \frac{\partial M(a \cdot b \cdot c \cdot \alpha \cdot \beta \cdot \gamma)}{\partial \gamma} \\
&= -2 \sum_{i=1}^n \left(\left(m_i - \frac{\left(a - c \frac{\alpha}{b \sqrt{(\gamma)^2 + 1}} \right) (1 - e^{-bt_i}) + c(\operatorname{arsinh}(\alpha t_i + \gamma) - \operatorname{arsinh}(\gamma))}{1 + \beta e^{-bt_i}} \right) \right. \\
& \quad \times \left. \frac{\left(c \left(\frac{1}{\sqrt{(\alpha t_i + \gamma)^2 + 1}} - \frac{1}{\sqrt{(\gamma)^2 + 1}} \right) + \frac{c\alpha\gamma(1 - e^{-bt_i})}{b(\gamma^2 + 1)^{\frac{3}{2}}} \right)}{1 + \beta e^{-bt_i}} \right) = 0 \quad (4.7)
\end{aligned}$$

第二節 實際失效數據集

一、資料集 一

資料源自於(Ohba, 1984b)，根據 Ohba 的測試資料。系統的測試時間為 21 天，共發現 46 個錯誤，詳細的故障紀錄如表 8 所示。

表 8 在線資料紀錄軟體測試資料

天	累積錯誤數目	天	累積錯誤數目	天	累積錯誤數目
1	2	8	12	15	31
2	3	9	19	16	37
3	4	10	21	17	38
4	5	11	22	18	41
5	7	12	24	19	42
6	9	13	26	20	45
7	11	14	30	21	46

資料來源：(Ohba, 1984b)

二、資料集 二

資料源自於(Musa, 1987)，根據一個即時指揮與控制系統，其數具來源為 Bell Laboratories 所開發之軟體。系統的測試時間為 25 小時，共發現 136 個錯誤，詳細的故障紀錄如表 9 所示。

表 9 即時指揮與控制系統

時	累積錯誤數目	時	累積錯誤數目	時	累積錯誤數目
1	27	10	93	19	128
2	43	11	97	20	129
3	54	12	104	21	131
4	64	13	106	22	132
5	75	14	111	23	134
6	83	15	116	24	135
7	84	16	122	25	136
8	89	17	122		
9	92	18	127		

資料來源：(Musa, 1987)

三、資料集 三

資料源自於(Wood,1996)，根據 Tandem Computers 的產品收集而來，此軟體於測試與修正的過程中，共發佈四個階段版本的軟體失效數據，而本研究僅採用第一階段版本的軟體失效數據。第一個階段版本的系統測試時間為期 20 週，共發現 100 個錯誤，詳細的故障紀錄如表 10 所示。

表 10 Tandem Computers 失效數據

週	累積錯誤數目	週	累積錯誤數目	週	累積錯誤數目
1	16	8	58	15	96
2	24	9	69	16	98
3	27	10	75	17	99
4	33	11	81	18	100
5	41	12	86	19	100
6	49	13	90	20	100
7	54	14	93		

資料來源：(Wood,1996)

四、資料集 四

資料源自於(Jeske & Zhang, 2005)，根據一個無線網路交換中心(Wireless Network Switching Center)，其主要服務為配置音頻和傳遞訊息至相關的無線資源和處理器中。系統生命週期為 13 個月，共發現 115 個錯誤，詳細的故障紀錄如表 11 所示。

表 11 無線網路交換中心失效數據

月	累積錯誤數目	月	累積錯誤數目	月	累積錯誤數目
1	7	6	51	11	111
2	10	7	58	12	115
3	24	8	77	13	115
4	32	9	94		
5	43	10	100		

資料來源：(Jeske & Zhang, 2005)

五、資料集 五

資料源自於(Tohma, Yamano, Ohba & Jacoby, 1991)一個用於即時控制的應用程式，此軟體系統是由 870,000 行程式碼所構成(Lyu, 1996)。累計測試時間為 109 天，共發現 535 個錯誤，詳細的故障紀錄如表 12 所示。

表 12 即時控制的應用程式失效數據

天	累積錯誤數目	天	累積錯誤數目	天	累積錯誤數目
1	4	38	287	75	497
2	4	39	294	76	508
3	11	40	306	77	509
4	21	41	318	78	509
5	34	42	333	79	511
6	42	43	347	80	513
7	55	44	354	81	517
8	59	45	363	82	518
9	66	46	374	83	518
10	74	47	379	84	522
11	75	48	386	85	523
12	81	49	393	86	524
13	94	50	407	87	524
14	101	51	420	88	526
15	110	52	434	89	526
16	118	53	445	90	526
17	123	54	447	91	527
18	133	55	451	92	528
19	140	56	455	93	528
20	151	57	458	94	528
21	156	58	464	95	528
22	164	59	470	96	528
23	177	60	473	97	529
24	186	61	473	98	529
25	193	62	473	99	530
26	200	63	476	100	530
27	205	64	476	101	530
28	212	65	480	102	530
29	218	66	480	103	530
30	224	67	481	104	532
31	228	68	483	105	532
32	240	69	483	106	533
33	246	70	484	107	533
34	253	71	486	108	535
35	261	72	491	109	535
36	272	73	494		
37	278	74	496		

資料來源：(Lyu, 1996)

六、資料集 六

資料源自於(Tohma, Tokunaga, Nagase, & Murata, 1989)一個即時監視與控制系統，此軟體系統由 200,000 行程式碼組成(Lyu, 1996)。累計測試時間為 111 天，共發現 481 個錯誤，詳細的故障紀錄如表 13 所示。

表 13 即時監視與控制系統失效數據

天	累積錯誤數目	天	累積錯誤數目	天	累積錯誤數目
1	5	38	324	75	469
2	10	39	331	76	469
3	15	40	346	77	470
4	20	41	367	78	472
5	26	42	375	79	472
6	34	43	381	80	472
7	36	44	401	81	473
8	43	45	411	82	473
9	47	46	414	83	473
10	49	47	417	84	473
11	80	48	425	85	473
12	84	49	430	86	473
13	108	50	431	87	475
14	157	51	433	88	475
15	171	52	435	89	475
16	183	53	437	90	475
17	191	54	444	91	475
18	200	55	446	92	475
19	204	56	446	93	475
20	211	57	448	94	475
21	217	58	451	95	475
22	217	59	453	96	476
23	230	60	460	97	476
24	234	61	463	98	476
25	236	62	463	99	476
26	240	63	464	100	477
27	243	64	464	101	477
28	252	65	465	102	477
29	254	66	465	103	478
30	259	67	465	104	478
31	263	68	466	105	478
32	264	69	467	106	479
33	268	70	467	107	479
34	271	71	467	108	479
35	277	72	468	109	480
36	290	73	469	110	480
37	309	74	469	111	481

資料來源：(Lyu, 1996)

七、資料集 七

資料源自於(Tohma et al., 1989)一個鐵路聯鎖(Interlocking)系統，此軟體系統約控制 14,500 個號誌(Lyu, 1996)。累計測試時間為 199 天，共發現 55 個錯誤，詳細的故障紀錄如表 14 所示。

表 14 鐵路聯鎖系統失效數據

天	累積錯誤數目	天	累積錯誤數目	天	累積錯誤數目
1	0	38	3	75	20
2	0	39	3	76	20
3	0	40	4	77	21
4	0	41	4	78	21
5	0	42	4	79	21
6	0	43	4	80	21
7	0	44	5	81	21
8	0	45	5	82	21
9	0	46	5	83	21
10	0	47	5	84	21
11	0	48	5	85	21
12	0	49	5	86	22
13	0	50	5	87	24
14	0	51	5	88	24
15	2	52	5	89	24
16	2	53	5	90	25
17	3	54	5	91	25
18	3	55	6	92	25
19	3	56	6	93	25
20	3	57	9	94	25
21	3	58	9	95	25
22	3	59	9	96	27
23	3	60	11	97	27
24	3	61	13	98	27
25	3	62	13	99	27
26	3	63	15	100	27
27	3	64	16	101	27
28	3	65	16	102	27
29	3	66	16	103	27
30	3	67	16	104	27
31	3	68	18	105	28
32	3	69	20	106	28
33	3	70	20	107	28
34	3	71	20	108	28
35	3	72	20	109	28
36	3	73	20	110	28
37	3	74	20	111	28

資料來源：(Lyu, 1996)

續表 14 鐵路聯鎖系統失效數據

天	累積錯誤數目	天	累積錯誤數目	天	累積錯誤數目
112	29	142	42	172	53
113	29	143	45	173	53
114	29	144	46	174	53
115	30	145	47	175	53
116	30	146	47	176	53
117	30	147	48	177	53
118	30	148	49	178	54
119	30	149	50	179	54
120	30	150	50	180	54
121	30	151	50	181	54
122	30	152	50	182	54
123	30	153	50	183	55
124	30	154	51	184	55
125	32	155	51	185	55
126	34	156	51	186	55
127	34	157	51	187	55
128	35	158	51	188	55
129	36	159	51	189	55
130	36	160	52	190	55
131	36	161	52	191	55
132	36	162	53	192	55
133	37	163	53	193	55
134	38	164	53	194	55
135	39	165	53	195	55
136	39	166	53	196	55
137	39	167	53	197	55
138	39	168	53	198	55
139	39	169	53	199	55
140	40	170	53		
141	42	171	53		

資料來源：(Lyu, 1996)

第三節 模型比較與測試

本研究利用七組真實失效數據集進行模型測試，並且利用文獻探討中所舉例的十個以往學者所提出的模型進行比較，確認模型是否能夠在實際應用的情況下有效的預估軟體的失效行為。

其中各模型所使用的參數根據其所提出之論文內文而有不同的定義，本研究忠實呈現所引用模型的原先變數定義，因此某些變數雖然符號相同，但所代表的意義卻不同，例如： α 於大多數模型中代表錯誤產生率，但在 P-EID model 中則代表初始錯誤數。

一、資料集一

表 15 為藉由資料集一(Data_set 1)估算出的各個現有模型與本研究所提模型之參數數值，分別帶入各自的均值函數中，即可得出各模型於各個時間點的預測累積失效數量。圖 10 至圖 20 為各模型與資料集一的配適度曲線，而圖 21 則是各模型的配適度曲線比較。

表 15 資料集一之各模型參數估計數值

模型\參數	a	b	c	α	β	γ
*Arsinh model	26.2149	0.2948	4.4874	0.5896	8.6329	-9.4518
G-O model	672.3853	0.0032	—	—	—	—
DSS model	77.2530	0.0966	—	—	—	—
ISS model	59.2854	0.1684	—	—	8.2782	—
Y-LID model	9.8759	0.1264	—	0.2860	—	—
Y-EID model	130.1228	0.0128	—	0.0392	—	—
H/D G-O model	675.9995	0.0032	2.9995	—	—	—
P-Z model	8.4885	0.1600	51.9257	0.2438	7.0901	—
P-N-Z model	51.4374	0.1724	—	0.0070	7.3871	—
P-EID model	—	0.1715	7.6477	53.5828	0.0047	—
Roy model	85.4626	0.0765	—	1.0868	0.0774	—

資料來源：本研究整理

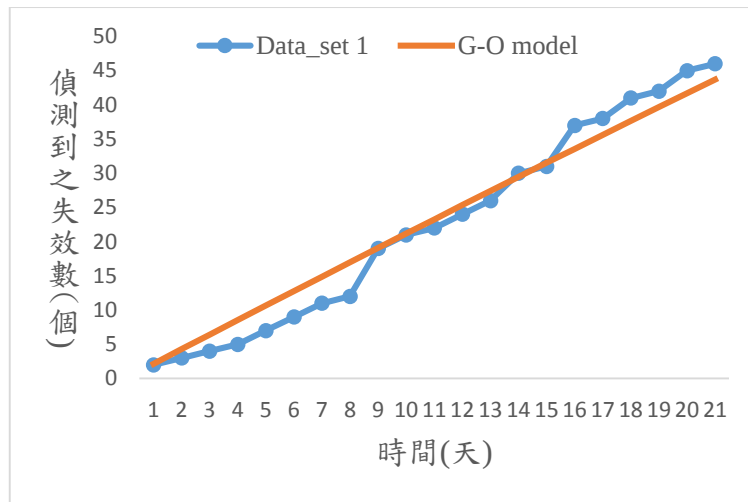


圖 10 資料集一與 G-O Model 之配適度曲線

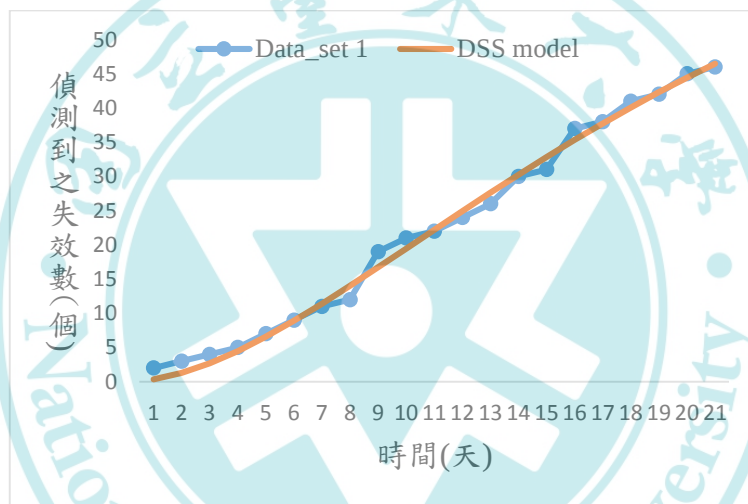


圖 11 資料集一與 DSS Model 之配適度曲線

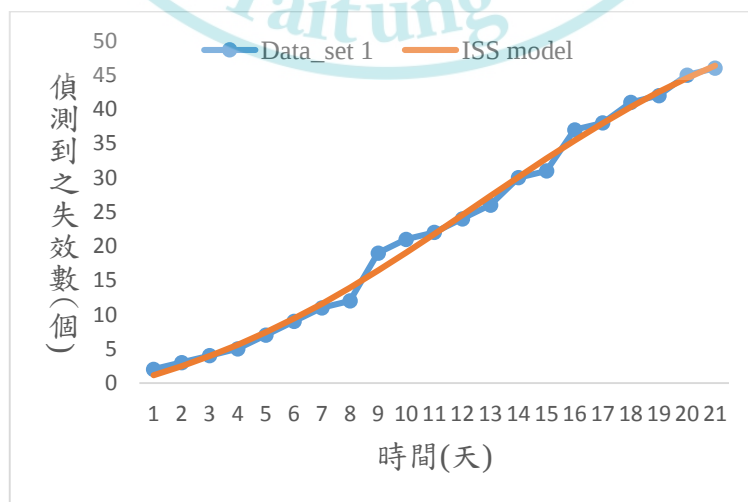


圖 12 資料集一與 ISS Model 之配適度曲線

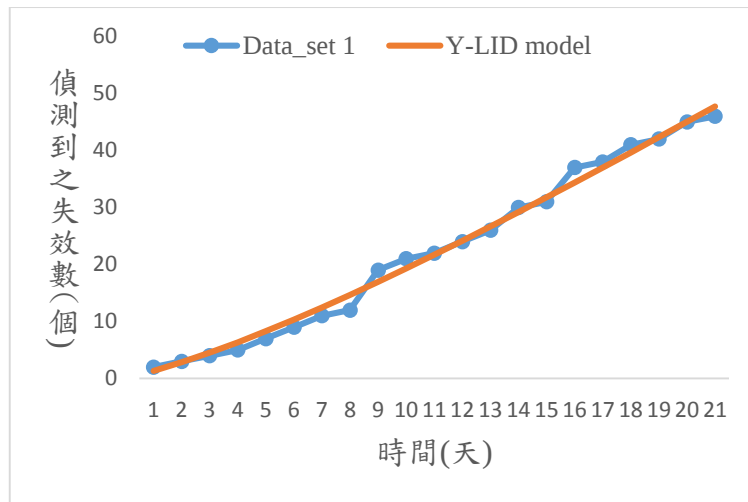


圖 13 資料集一與 Y-LID Model 之配適度曲線

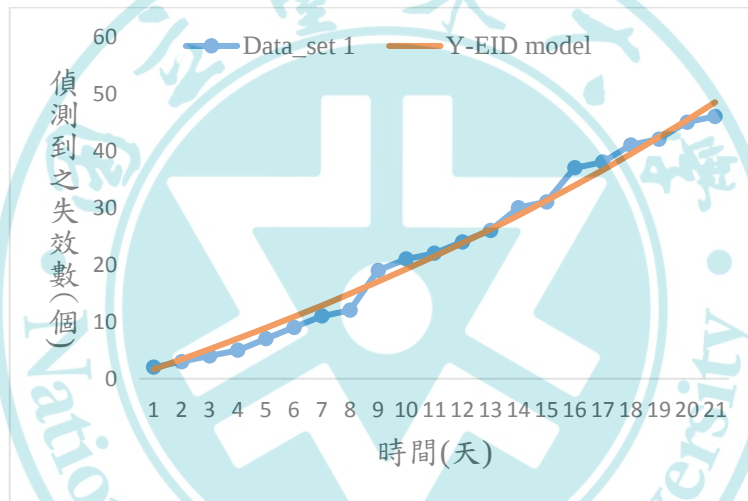


圖 14 資料集一與 Y-EID Model 之配適度曲線

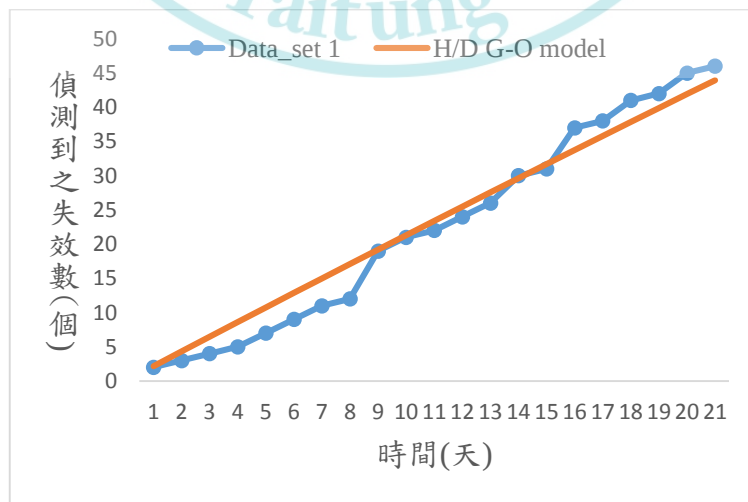


圖 15 資料集一與 H/D G-O Model 之配適度曲線

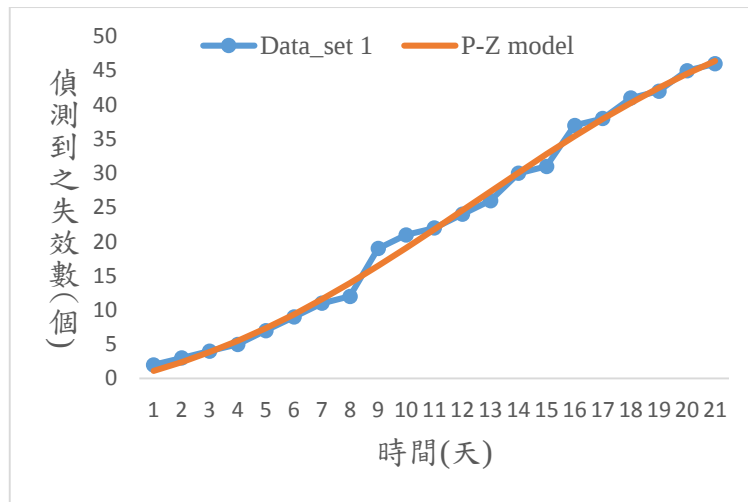


圖 16 資料集一與 P-Z Model 之配適度曲線

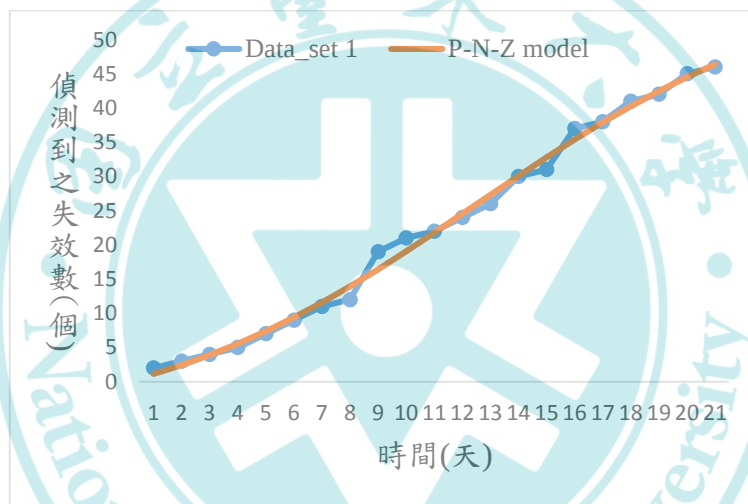


圖 17 資料集一與 P-N-Z Model 之配適度曲線

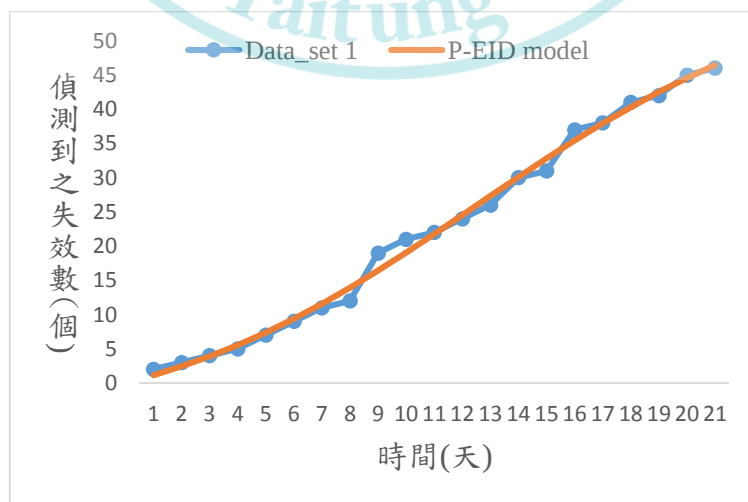


圖 18 資料集一與 P-EID Model 之配適度曲線

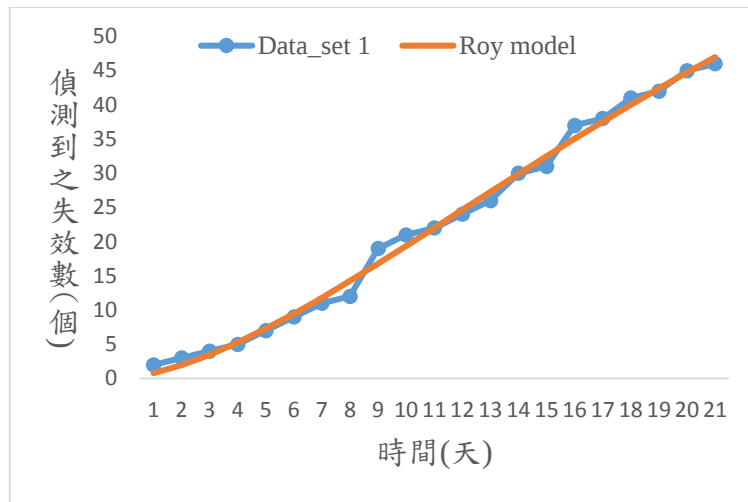


圖 19 資料集一與 Roy Model 之配適度曲線

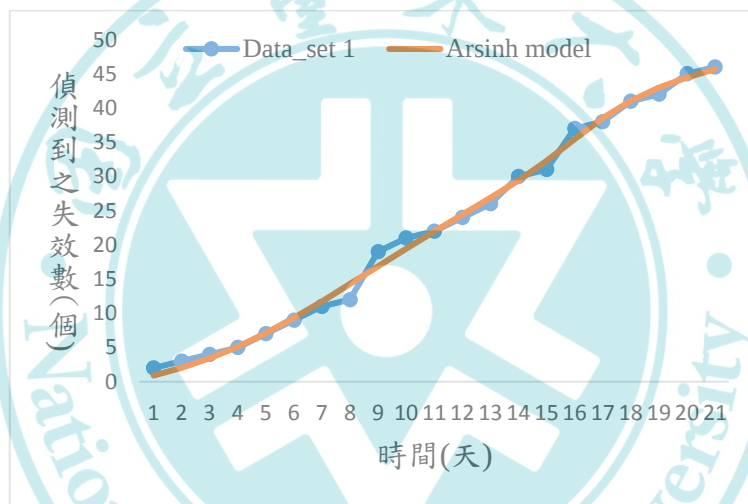


圖 20 資料集一與 Arsinh Model 之配適度曲線

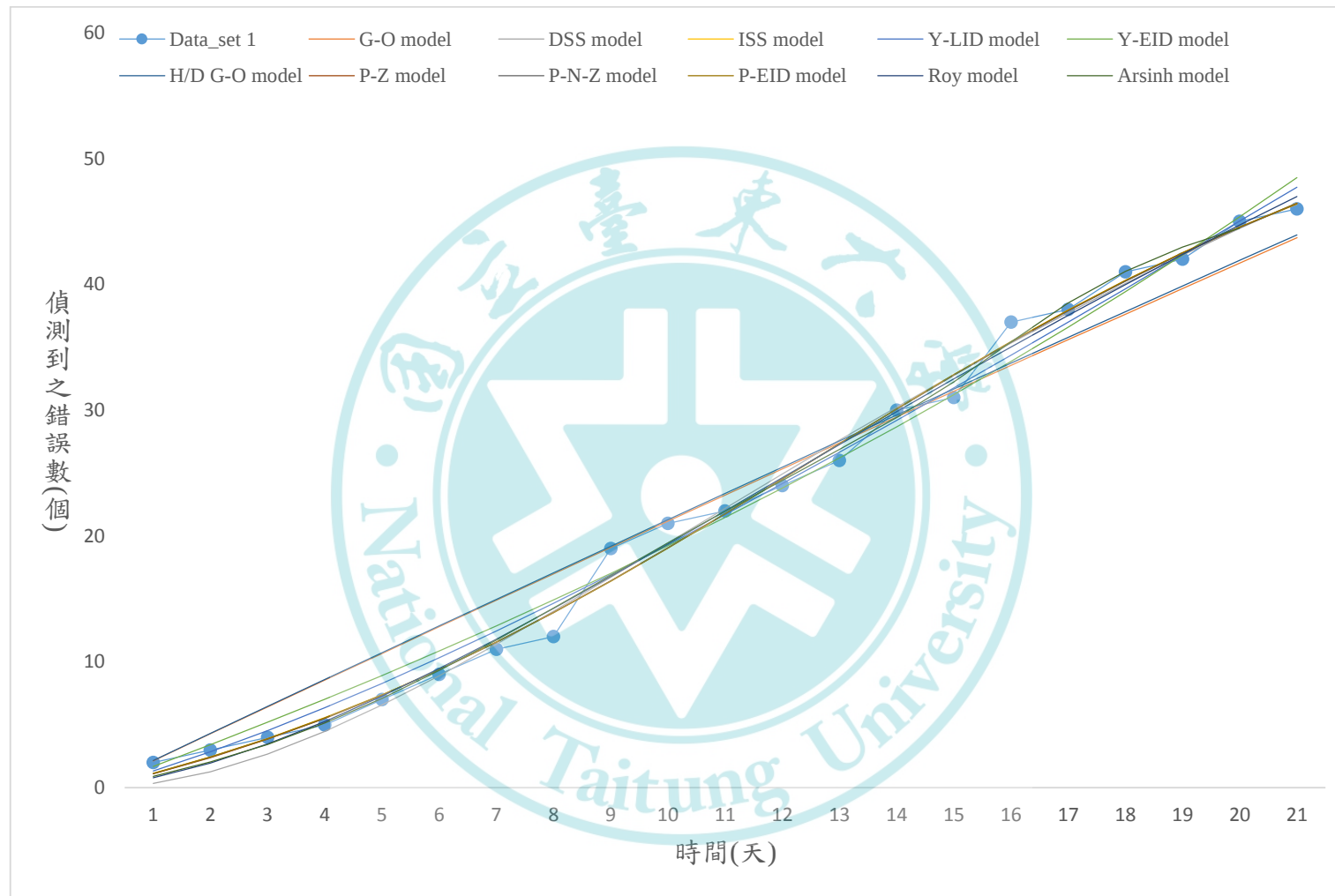


圖 21 資料集一與各模型之配適度曲線比較

表 16 各模型於資料集一的比較結果

模型\評估標準	MSE	R^2	SAE	PRV	RMSPE
Arsinh model	1.4727	0.9951	16.9638*	1.0479*	1.0508*
G-O model	7.6140	0.9679	46.2208	2.6371	2.6870
DSS model	1.6366	0.9931	21.0269	1.2228	1.2458
ISS model	1.3947*	0.9944	17.5069	1.1201	1.1204
Y-LID model	2.0447	0.9918	22.8684	1.3542	1.3564
Y-EID model	3.0422	0.9879	28.0299	1.6452	1.6542
H/D G-O model	7.8992	0.9685	46.0231	2.5845	2.6625
P-Z model	1.5655	0.9944	17.6016	1.1184	1.1191
P-N-Z model	1.4765	0.9944	17.5795	1.1195	1.1203
P-EID model	1.4767	0.9944	17.5717	1.1195	1.1197
Roy model	1.5848	0.9940	19.2528	1.1577	1.1605

資料來源：本研究整理

各模型於資料集一的模型評估標準數值如表 16 所示。由表 16 可得知，本研究所提之模型在資料集一中，除了 MSE 這項指標落後於 ISS Model 之外，其餘四個指標都表現得優於另外十個模型。

二、資料集二

表 17 為藉由資料集二(Data_set 2)估算出的各個現有模型與本研究所提模型之參數數值，分別帶入各自的均值函數中，即可得出各模型於各個時間點的預測累積失效數量。圖 22 至圖 32 為各模型與資料集二的配適度曲線，而圖 33 則是各模型的配適度曲線比較。

表 17 資料集二之各模型參數估計數值

模型\參數	a	b	c	α	β	γ
*Arsinh model	89.3622	0.3216	9.4318	0.5176	0.0002	-7.2851
G-O model	135.8571	0.1388	—	—	—	—
DSS model	124.6045	0.3573	—	—	—	—
ISS model	136.0000	0.1400	—	—	0.0001	—
Y-LID model	81.2969	0.3407	—	0.0332	—	—
Y-EID model	89.8551	0.2907	—	0.0208	—	—
H/D G-O model	135.8388	0.1388	1.0000	—	—	—
P-Z model	115.7765	0.7295	52.2741	0.0575	0.247	—
P-N-Z model	128.5965	0.1484	—	0.0034	0.0001	—
P-EID model	—	0.1456	0.0003	125.6327	0.0055	—
Roy model	113.6527	0.6047	—	1.4835	0.0567	—

資料來源：本研究整理

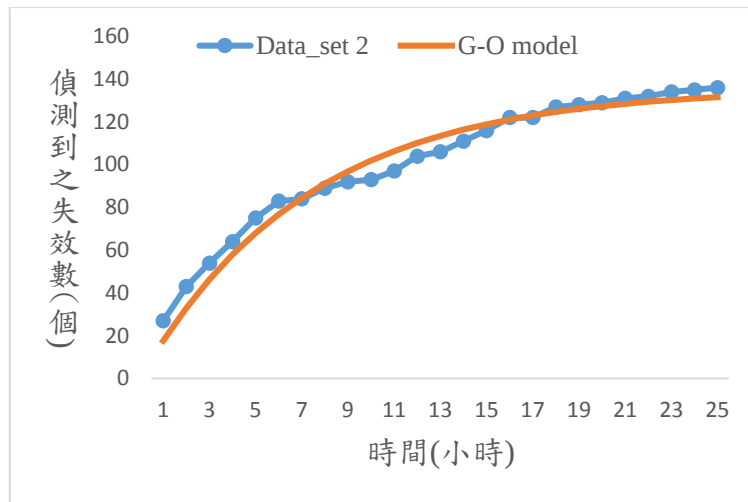


圖 22 資料集二與 G-O Model 之配適度曲線

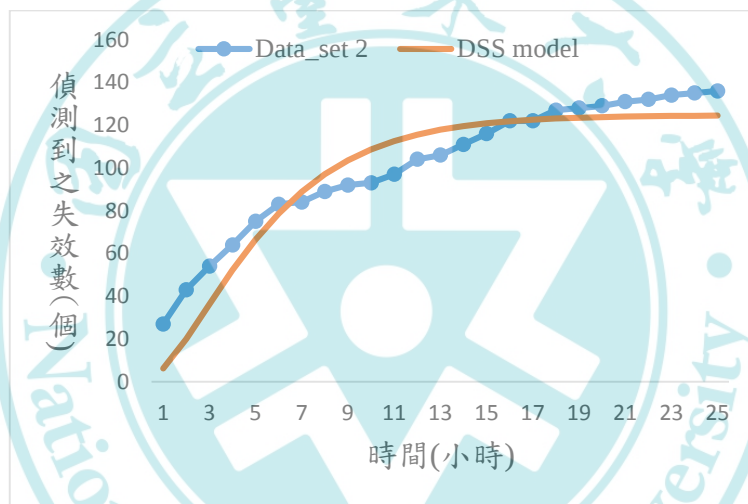


圖 23 資料集二與 DSS Model 之配適度曲線

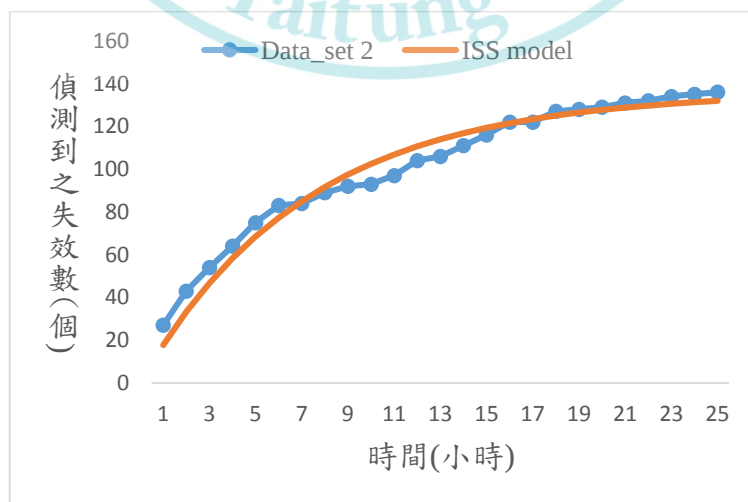


圖 24 資料集二與 ISS Model 之配適度曲線

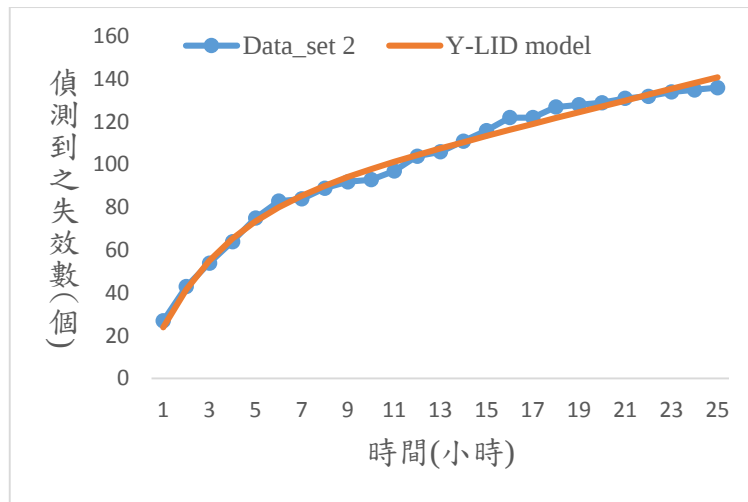


圖 25 資料集二與 Y-LID Model 之配適度曲線

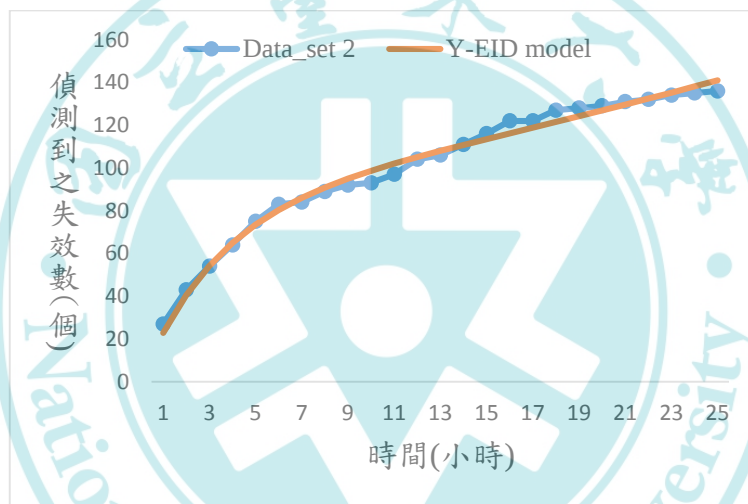


圖 26 資料集二與 Y-EID Model 之配適度曲線

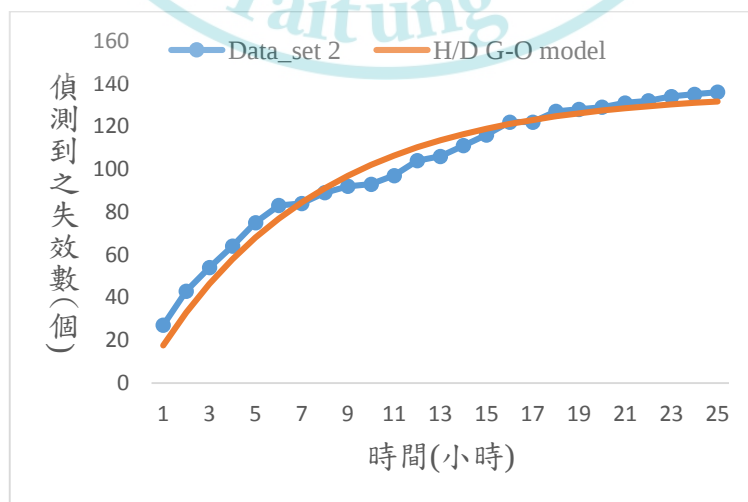


圖 27 資料集二與 H/D G-O Model 之配適度曲線

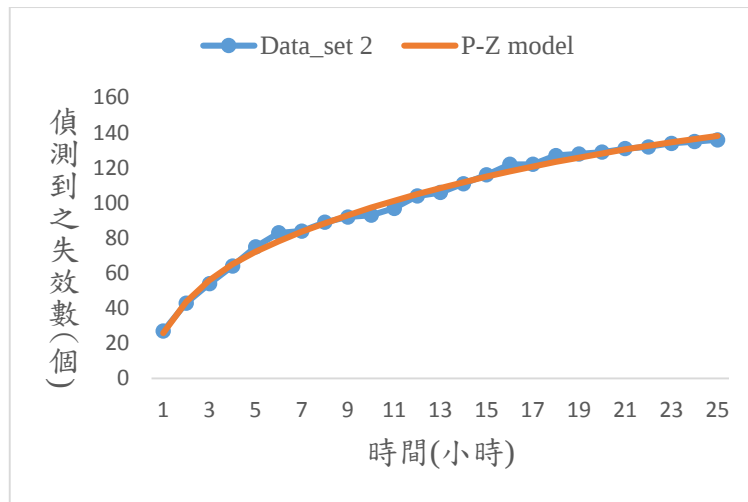


圖 28 資料集二與 P-Z Model 之配適度曲線

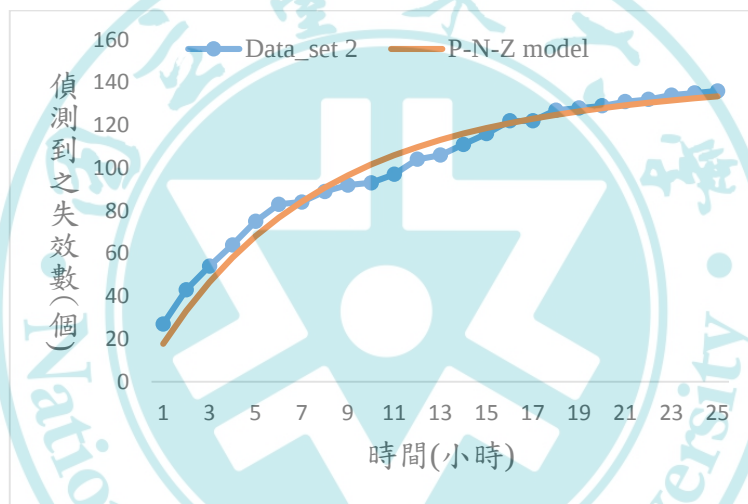


圖 29 資料集二與 P-N-Z Model 之配適度曲線

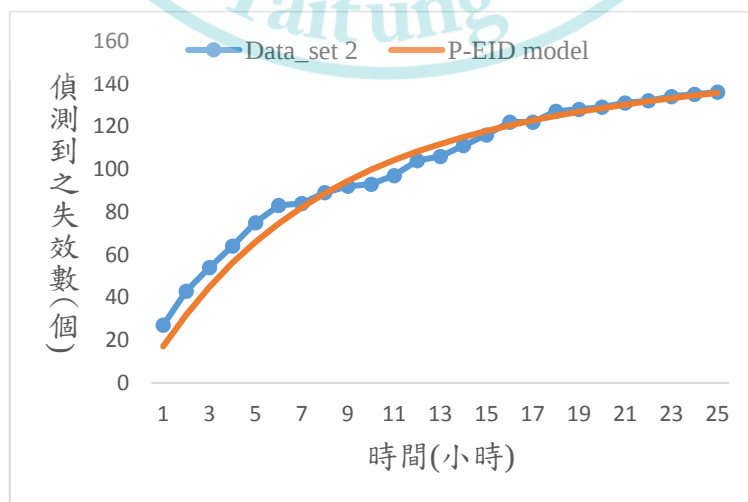


圖 30 資料集二與 P-EID Model 之配適度曲線

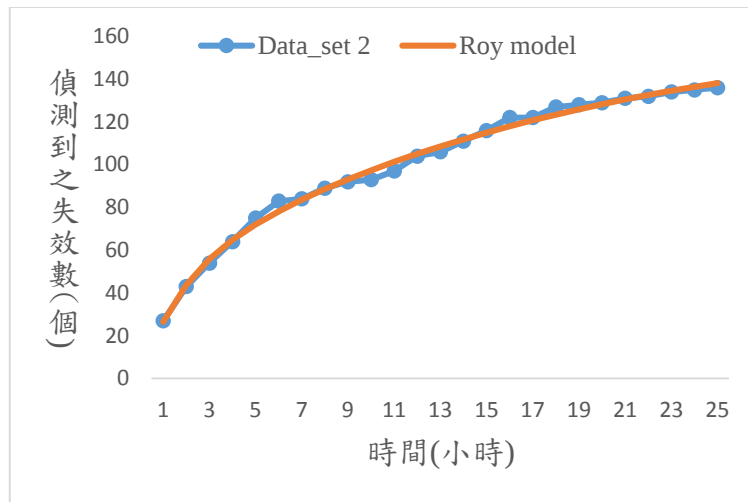


圖 31 資料集二與 Roy Model 之配適度曲線

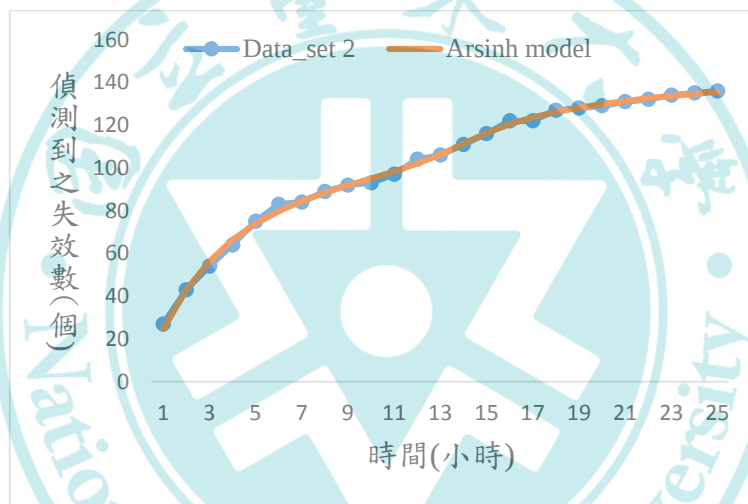


圖 32 資料集二與 Arsinh Model 之配適度曲線

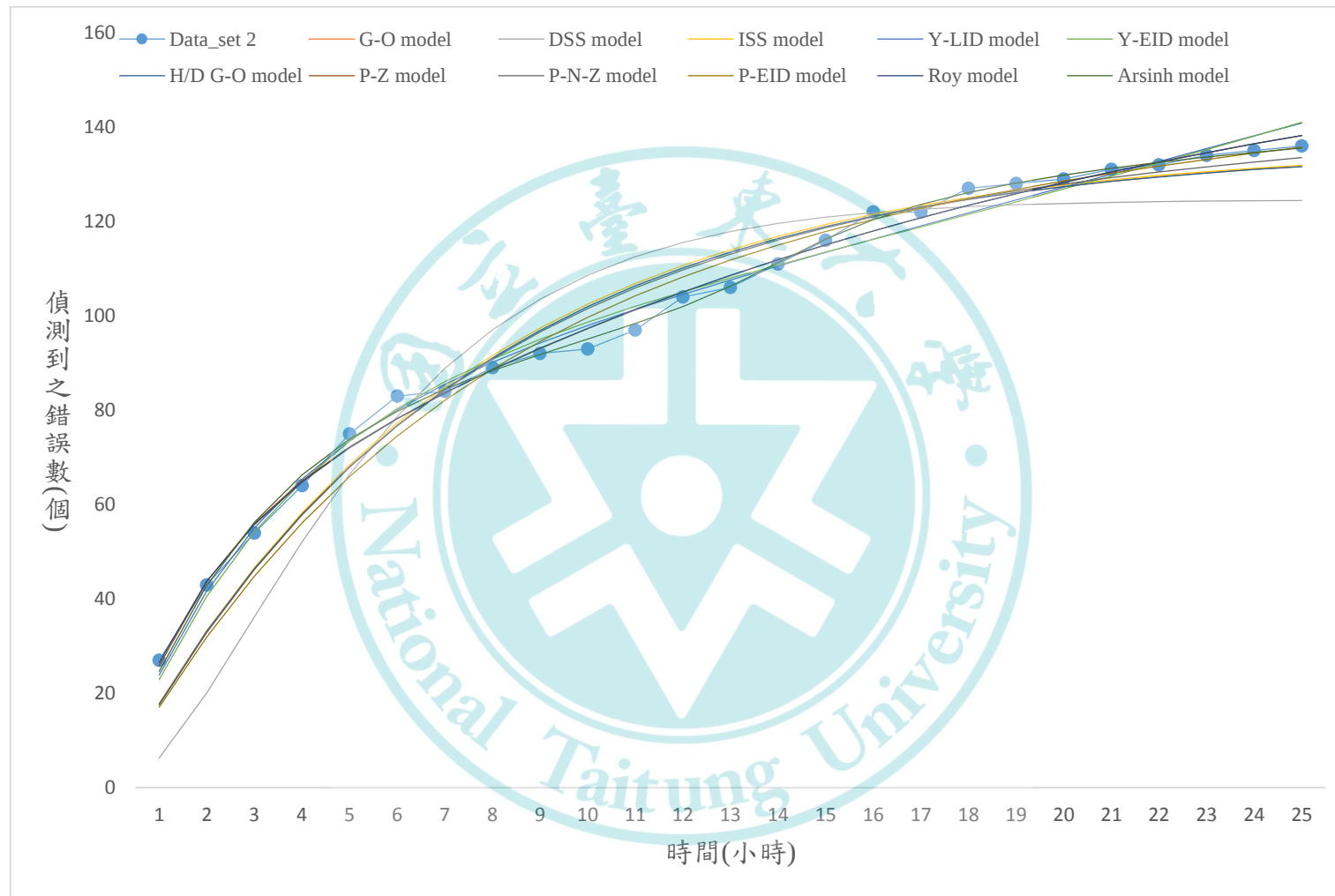


圖 33 資料集二與各模型之配適度曲線比較

表 18 各模型於資料集二的比較結果

模型\評估標準	MSE	R^2	SAE	PRV	RMSPE
Arsinh model	2.4063	0.9980*	25.8878*	1.3798*	1.3802*
G-O model	33.8114	0.9658	119.1109	5.6235	5.6896
DSS model	134.5736	0.8641	239.5801	11.1431	11.3479
ISS model	35.5391	0.9657	118.7548	5.6875	5.7069
Y-LID model	9.4335	0.9909	60.9045	2.9378	2.9405
Y-EID model	11.3422	0.9890	66.6980	3.2201	3.2243
H/D G-O model	35.3379	0.9659	119.1342	5.6206	5.6887
P-Z model	6.3774	0.9944	45.1648	2.3053	2.3053
P-N-Z model	32.8109	0.9697	107.8942	5.3136	5.3564
P-EID model	33.7168	0.9689	98.8205	5.2632	5.4250
Roy model	6.1242	0.9944	44.6621	2.3149	2.3149

資料來源：本研究整理

各模型於資料集二的模型評估標準數值如表 18 所示。由表 18 可得知，本研究所提之模型在資料集二中，五項評估指標皆優於另外十個模型。

三、資料集三

表 19 為藉由資料集三(Data_set 3)估算出的各個現有模型與本研究所提模型之參數數值，分別帶入各自的均值函數中，即可得出各模型於各個時間點的預測累積失效數量。圖 34 至圖 44 為各模型與資料集三的配適度曲線，而圖 45 則是各模型的配適度曲線比較。

表 19 資料集三之各模型參數估計數值

模型\參數	a	b	c	α	β	γ
*Arsinh model	46.1487	0.3013	12.1548	0.6026	0.0315	-5.4203
G-O model	130.2017	0.0832	—	—	—	—
DSS model	103.9839	0.2654	—	—	—	—
ISS model	110.8298	0.1721	—	—	1.2046	—
Y-LID model	129.4503	0.0823	—	0.0004	—	—
Y-EID model	135.3125	0.0778	—	0.0001	—	—
H/D G-O model	130.2012	0.0832	0.1094	—	—	—
P-Z model	91.2185	0.1683	110.7437	0.0002	1.1111	—
P-N-Z model	118.7409	0.1220	—	0.0001	0.4377	—
P-EID model	—	0.1616	1.3715	116.7883	0.0004	—
Roy model	62.2591	0.1689	—	1.8501	0.1688	—

資料來源：本研究整理

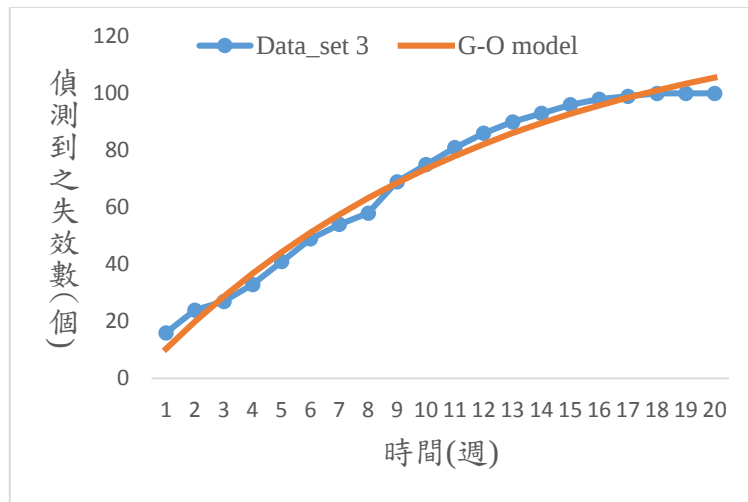


圖 34 資料集三與 G-O Model 之配適度曲線

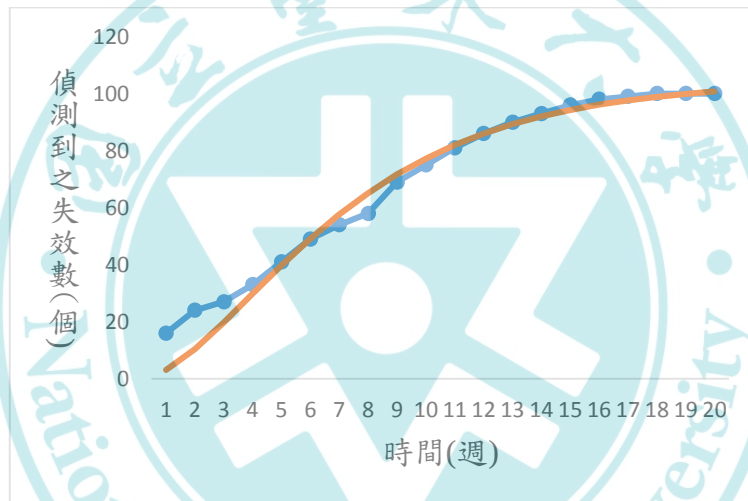


圖 35 資料集三與 DSS Model 之配適度曲線

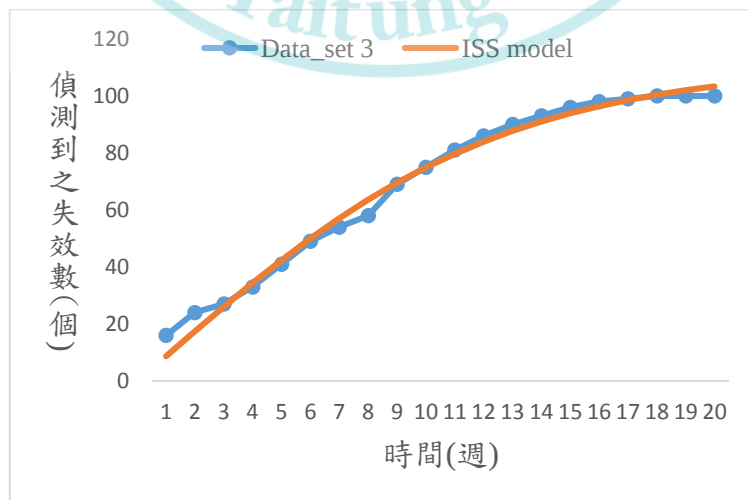


圖 36 資料集三與 ISS Model 之配適度曲線

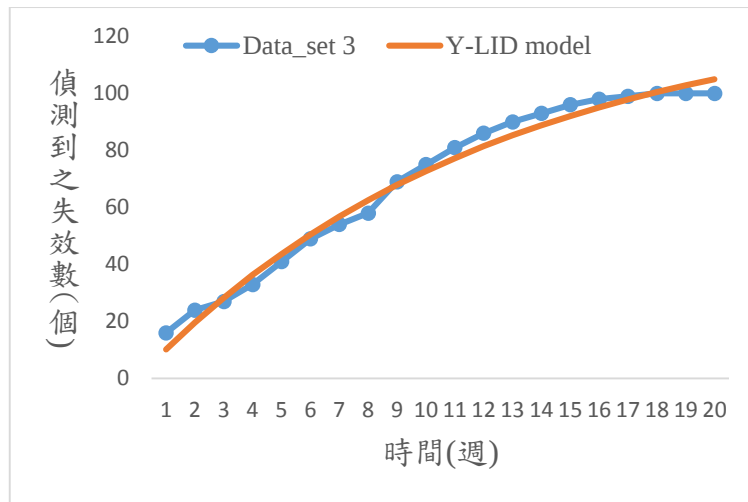


圖 37 資料集三與 Y-LID Model 之配適度曲線

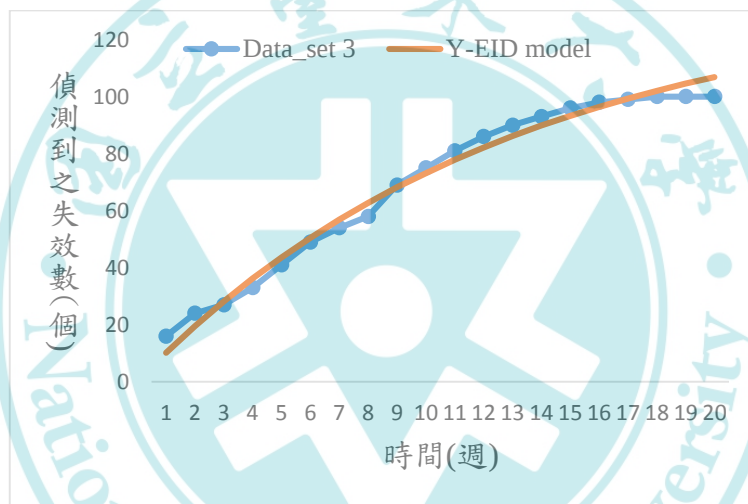


圖 38 資料集三與 Y-EID Model 之配適度曲線

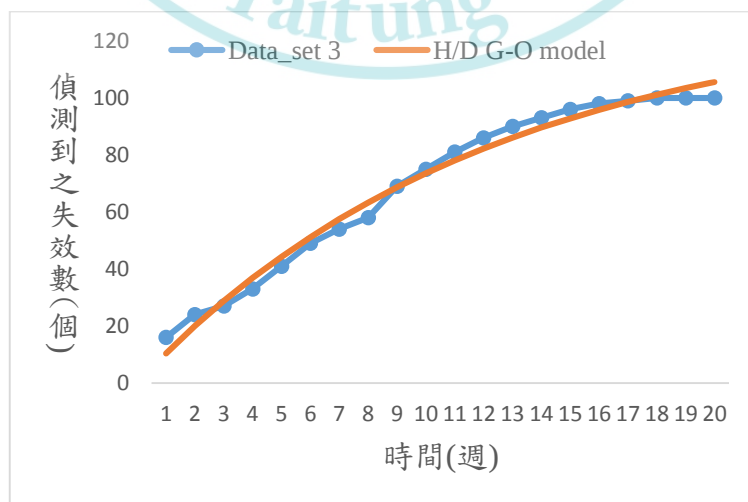


圖 39 資料集三與 H/D G-O Model 之配適度曲線

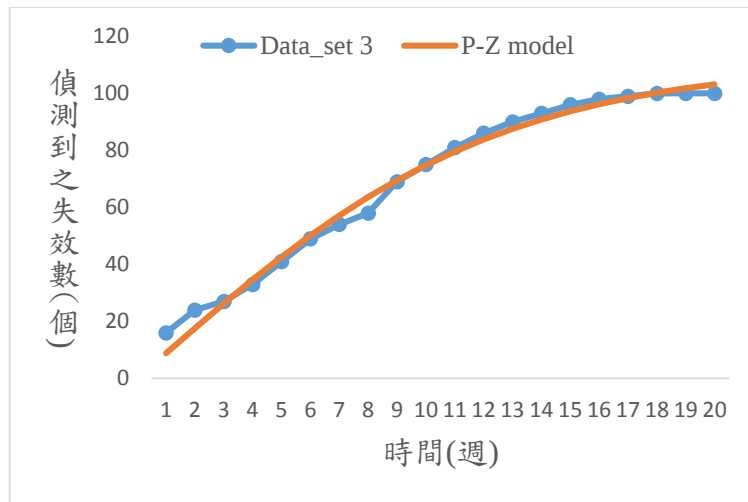


圖 40 資料集三與 P-Z Model 之配適度曲線

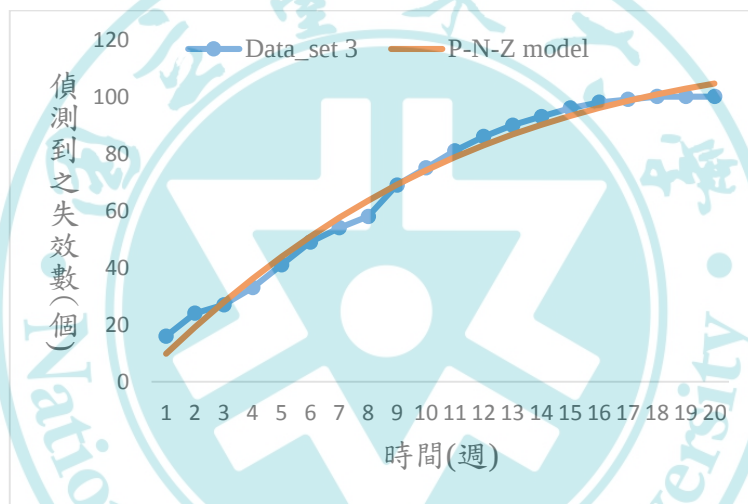


圖 41 資料集三與 P-N-Z Model 之配適度曲線

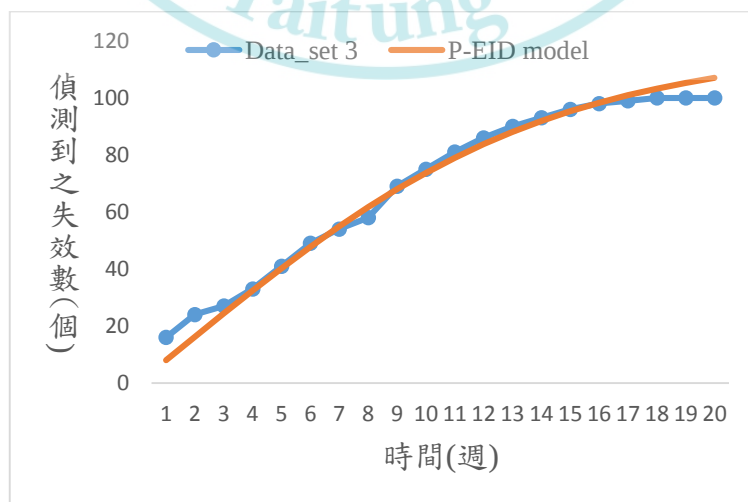


圖 42 資料集三與 P-EID Model 之配適度曲線

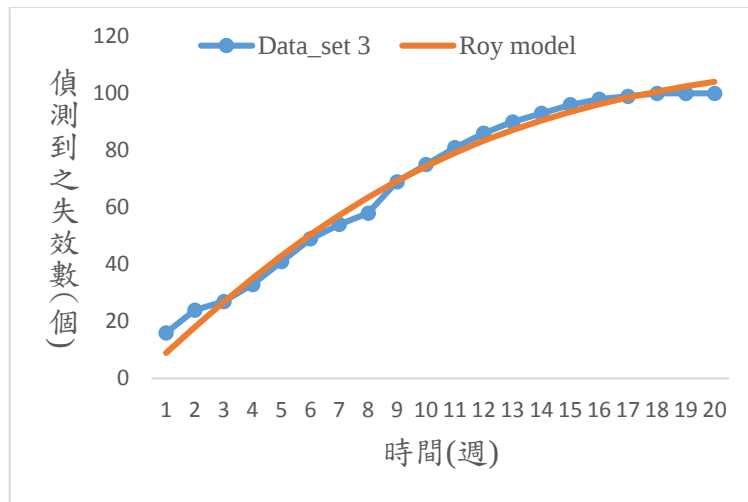


圖 43 資料集三與 Roy Model 之配適度曲線

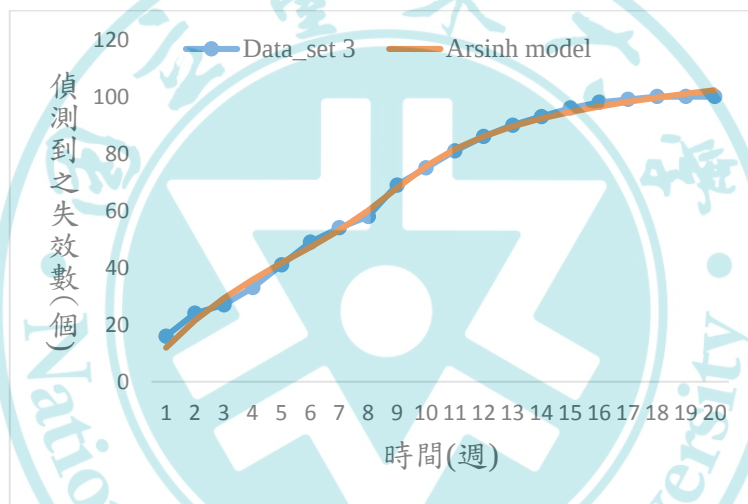


圖 44 資料集三與 Arsinh Model 之配適度曲線

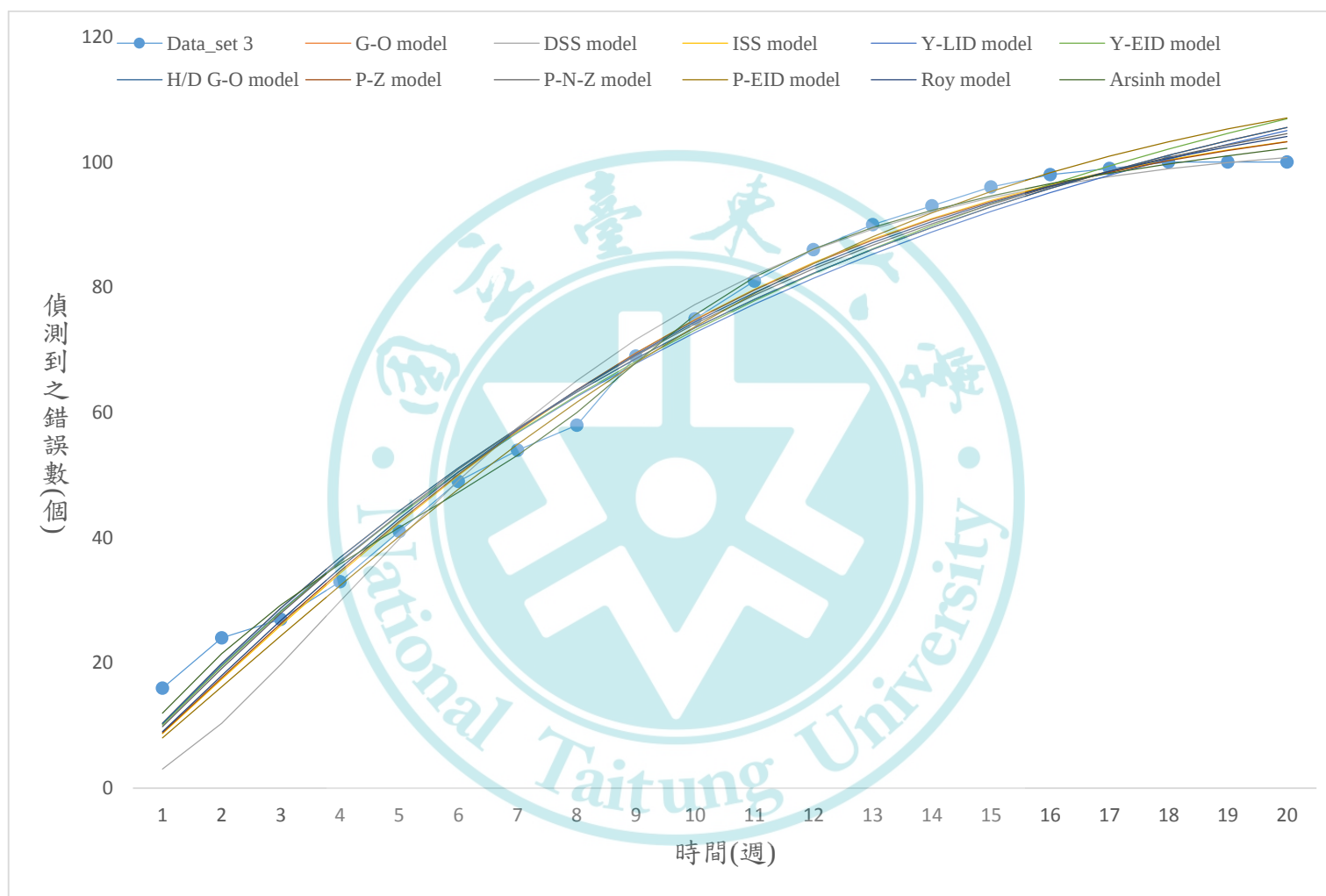


圖 45 資料集三與各模型之配適度曲線比較

表 20 各模型於資料集三的比較結果

模型\評估標準	MSE	R^2	SAE	PRV	RMSPE
Arsinh model	4.0436	0.9965*	27.4475*	1.7195*	1.7258*
G-O model	12.9082	0.9857	61.2524	3.4961	3.4969
DSS model	28.0627	0.9689	63.3548	4.9472	5.1459
ISS model	10.5639	0.9890	45.1959	3.0432	3.0728
Y-LID model	14.1817	0.9852	62.9810	3.4903	3.5586
Y-EID model	14.2139	0.9851	61.1884	3.5656	3.5662
H/D G-O model	13.6675	0.9857	61.2532	3.4961	3.4969
P-Z model	12.0299	0.9889	46.2993	3.0516	3.0803
P-N-Z model	12.6013	0.9876	54.5262	3.2551	3.2574
P-EID model	16.0810	0.9842	53.7754	3.6517	3.6785
Roy model	12.2113	0.9880	50.3113	3.1905	3.2059

資料來源：本研究整理

各模型於資料集三的模型評估標準數值如表 20 所示。由表 20 可得知，本研究所提之模型在資料集三中，五項評估指標皆優於另外十個模型。

四、資料集四

表 21 為藉由資料集四(Data_set 4)估算出的各個現有模型與本研究所提模型之參數數值，分別帶入各自的均值函數中，即可得出各模型於各個時間點的預測累積失效數量。圖 46 至圖 56 為各模型與資料集四的配適度曲線，而圖 57 則是各模型的配適度曲線比較。

表 21 資料集四之各模型參數估計數值

模型\參數	a	b	c	α	β	γ
*Arsinh model	33.7965	0.4308	18.3739	0.8612	1.0796	-6.4941
G-O model	985.7344	0.0100	—	—	—	—
DSS model	167.9492	0.1951	—	—	—	—
ISS model	134.4995	0.3363	—	—	8.9528	—
Y-LID model	10.8048	0.5345	—	0.9556	—	—
Y-EID model	887.4999	0.0098	—	0.0251	—	—
H/D G-O model	709.7818	0.0142	0.0082	—	—	—
P-Z model	10.0436	0.3398	123.2304	17.2846	8.9767	—
P-N-Z model	180.0929	0.2020	—	0.0001	4.6956	—
P-EID model	—	0.1019	0.0024	90.6083	0.0772	—
Roy model	166.9512	0.1809	—	1.0505	0.1808	—

資料來源：本研究整理

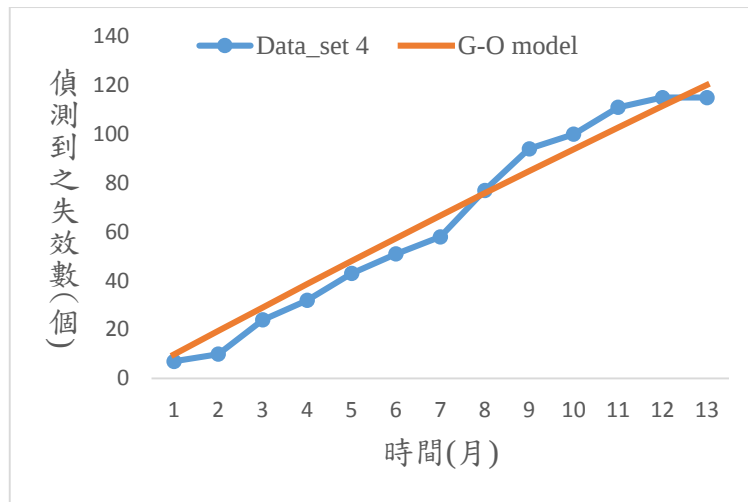


圖 46 資料集四與 G-O Model 之配適度曲線

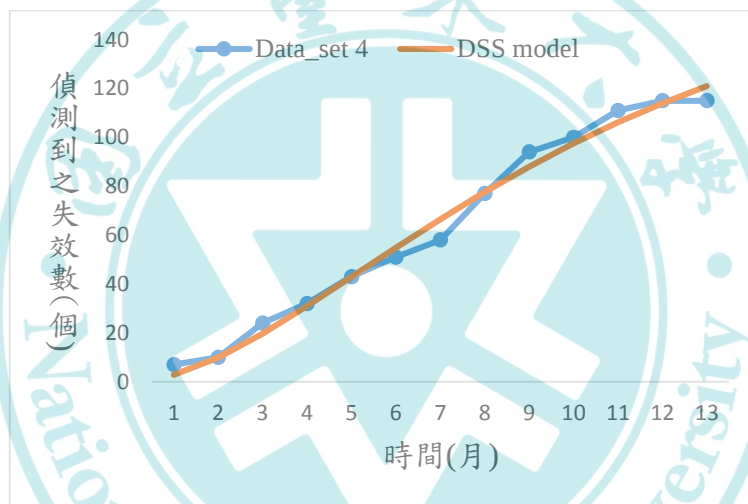


圖 47 資料集四與 DSS Model 之配適度曲線

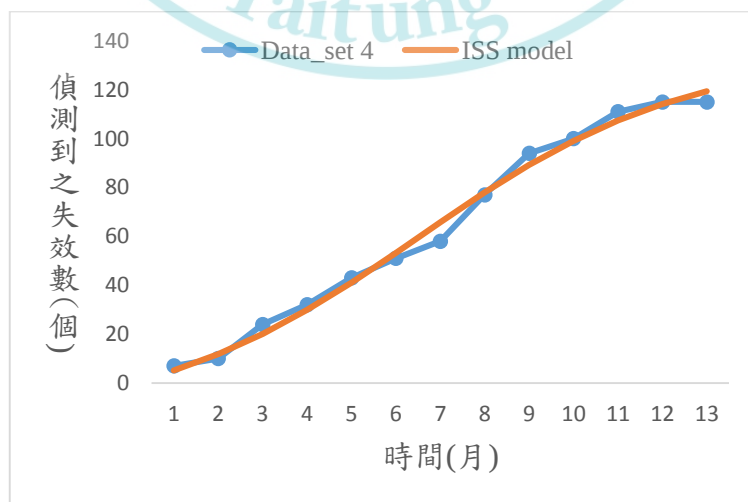


圖 48 資料集四與 ISS Model 之配適度曲線

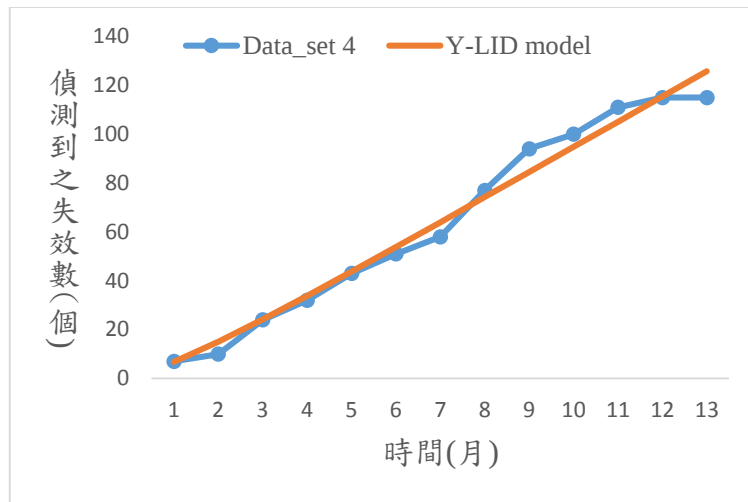


圖 49 資料集四與 Y-LID Model 之配適度曲線

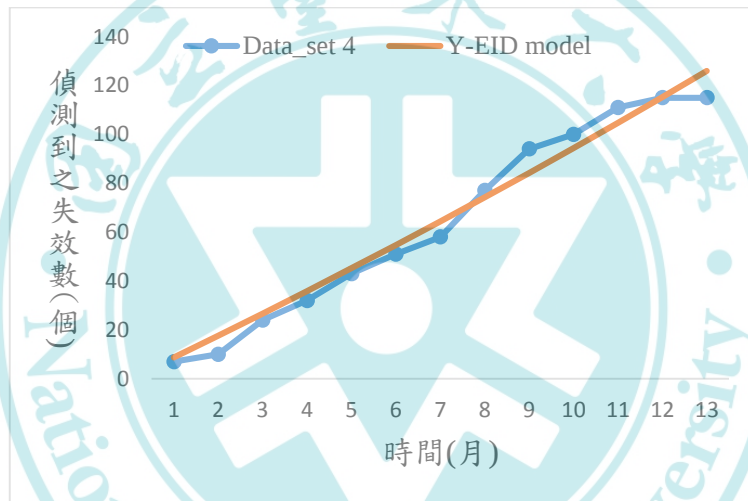


圖 50 資料集四與 Y-EID Model 之配適度曲線

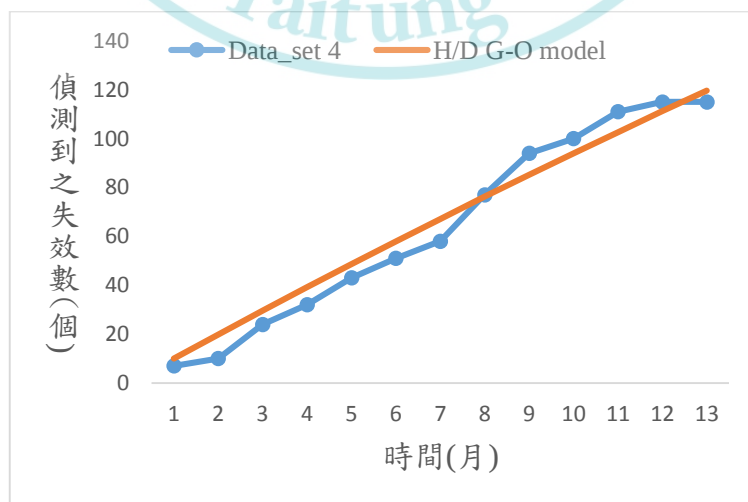


圖 51 資料集四與 H/D G-O Model 之配適度曲線

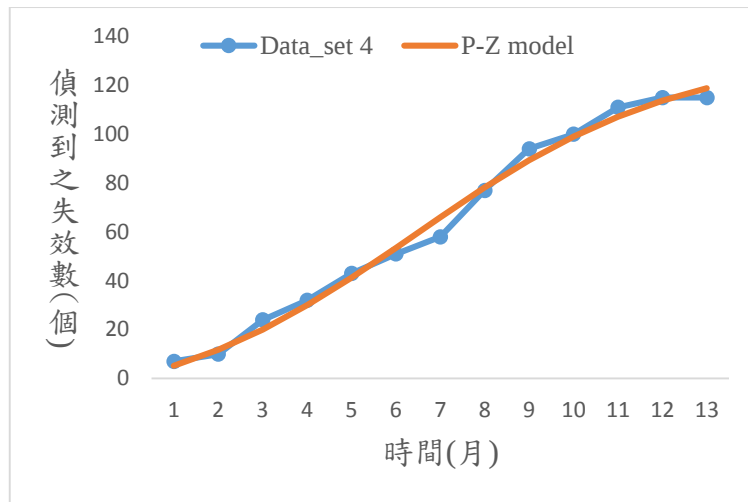


圖 52 資料集四與 P-Z Model 之配適度曲線

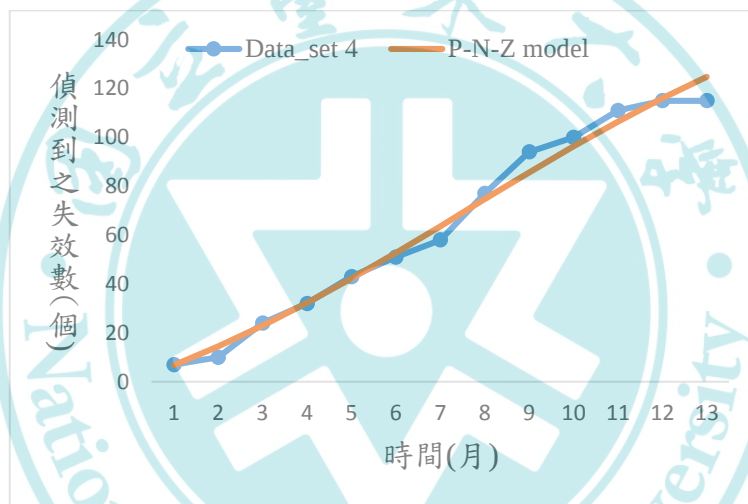


圖 53 資料集四與 P-N-Z Model 之配適度曲線

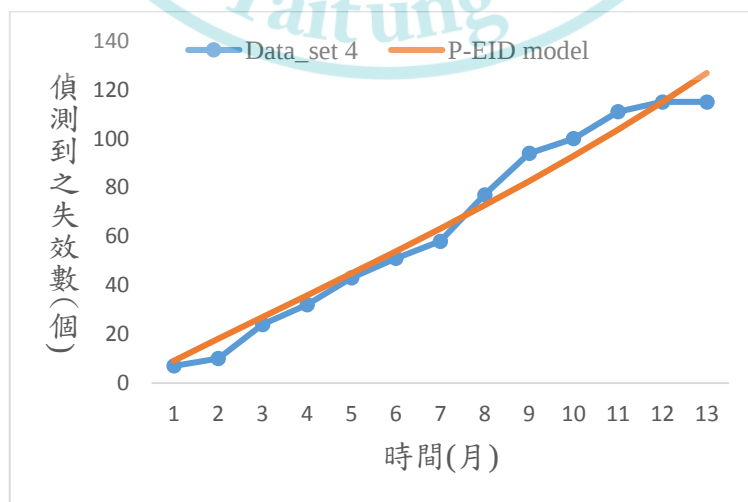


圖 54 資料集四與 P-EID Model 之配適度曲線

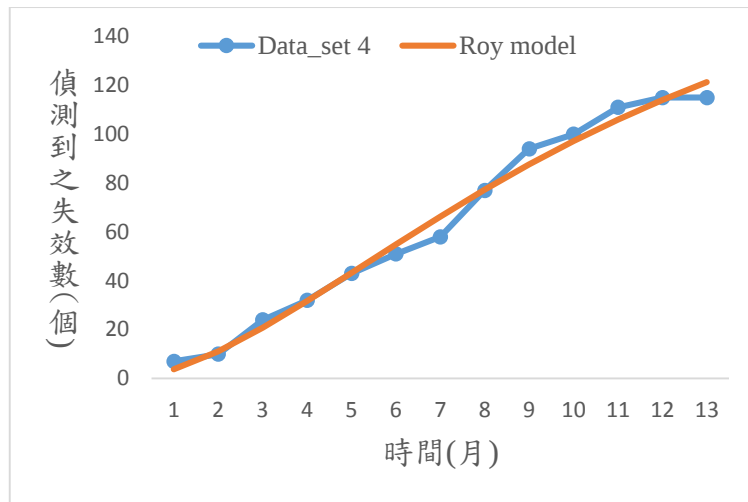


圖 55 資料集四與 Roy Model 之配適度曲線

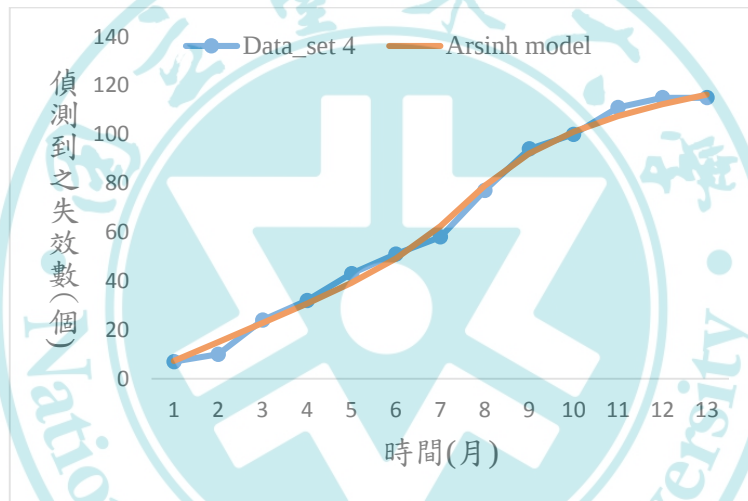


圖 56 資料集四與 Arsinh Model 之配適度曲線

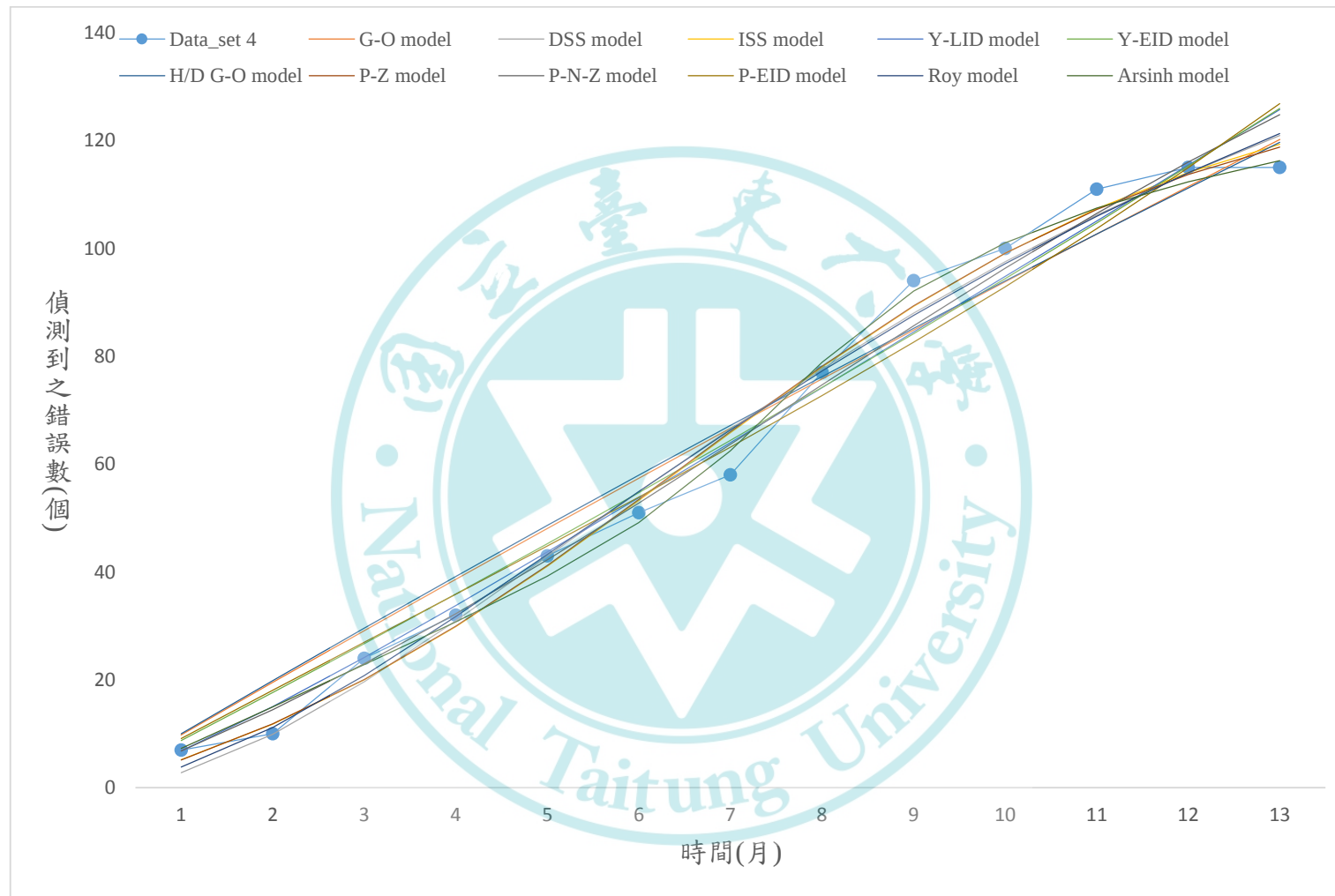


圖 57 資料集四與各模型之配適度曲線比較

表 22 各模型於資料集四的比較結果

模型\評估標準	MSE	R^2	SAE	PRV	RMSPE
Arsinh model	13.5880	0.9950*	30.0863*	2.8103*	2.8150*
G-O model	49.3330	0.9717	77.8187	6.5116	6.7086
DSS model	20.7414	0.9881	43.2176	4.3404	4.3589
ISS model	15.3196	0.9920	37.1993	3.5656	3.5724
Y-LID model	34.8427	0.9819	51.2441	5.3792	5.3877
Y-EID model	43.9597	0.9771	64.1241	5.9304	6.0432
H/D G-O model	57.5171	0.9701	79.9147	6.6449	6.9022
P-Z model	19.2432	0.9920	37.5102	3.5734	3.5811
P-N-Z model	29.2364	0.9863	43.8296	4.6792	4.6824
P-EID model	58.4001	0.9726	68.9636	6.5836	6.6155
Roy model	24.6769	0.9884	42.2255	4.2998	4.3019

資料來源：本研究整理

各模型於資料集四的模型評估標準數值如表 22 所示。由表 22 可得知，本研究所提之模型在資料集四中，五項評估指標皆優於另外十個模型。

五、資料集五

表 23 為藉由資料集五(Data_set 5)估算出的各個現有模型與本研究所提模型之參數數值，分別帶入各自的均值函數中，即可得出各模型於各個時間點的預測累積失效數量。圖 58 至圖 68 為各模型與資料集五的配適度曲線，而圖 69 則是各模型的配適度曲線比較。

表 23 資料集五之各模型參數估計數值

模型\參數	a	b	c	α	β	γ
*Arsinh model	217.9675	0.1046	54.0728	0.2092	3.2996	-9.1573
G-O model	728.9229	0.0146	—	—	—	—
DSS model	565.8381	0.049	—	—	—	—
ISS model	543.458	0.0595	—	—	5.5607	—
Y-LID model	757.6718	0.0136	—	0.0001	—	—
Y-EID model	744.975	0.0141	—	0.0001	—	—
H/D G-O model	709.7795	0.0153	0.7906	—	—	—
P-Z model	537.3851	0.0462	36.8391	0.0476	0.0843	—
P-N-Z model	570.0121	0.0443	—	0.0001	2.7353	—
P-EID model	—	0.0508	4.4124	567.4406	0.0001	—
Roy model	501.5451	0.044	—	1.145	0.0457	—

資料來源：本研究整理

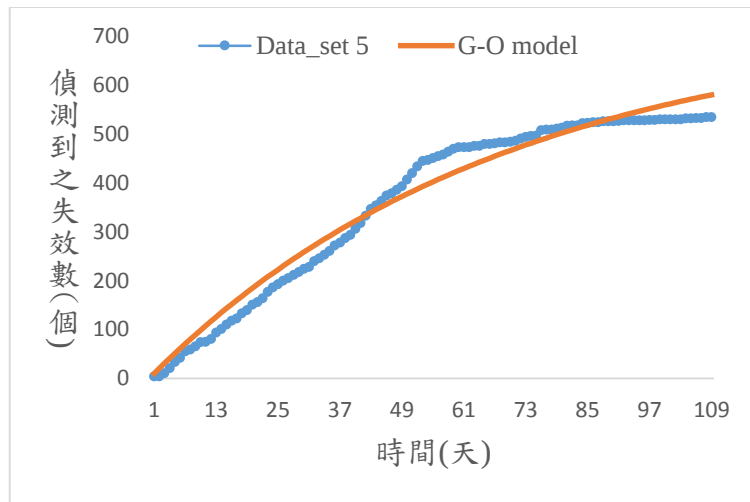


圖 58 資料集五與 G-O Model 之配適度曲線

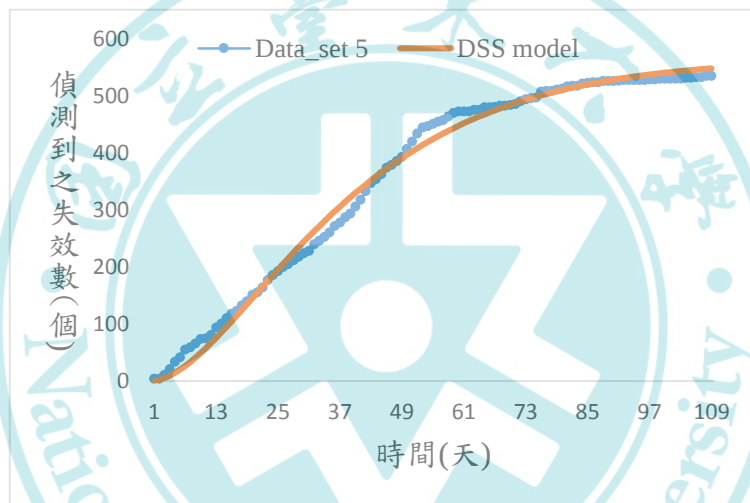


圖 59 資料集五與 DSS Model 之配適度曲線

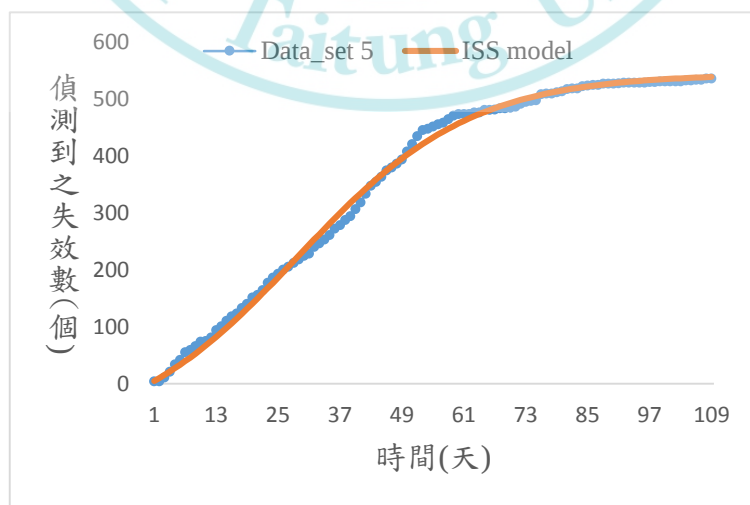


圖 60 資料集五與 ISS Model 之配適度曲線

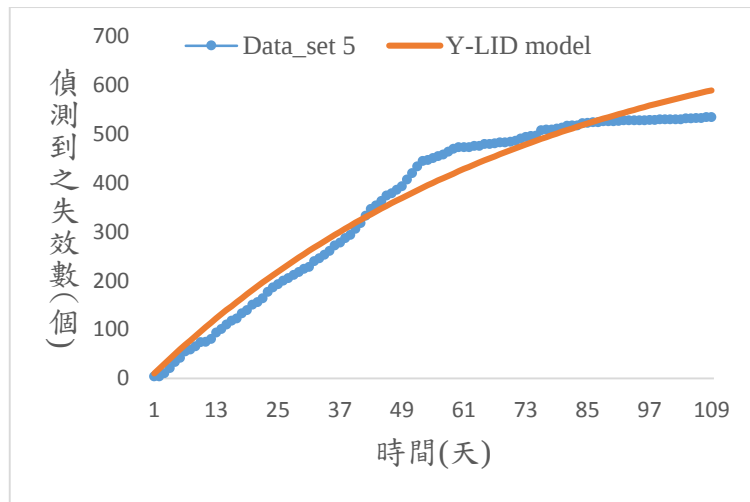


圖 61 資料集五與 Y-LID Model 之配適度曲線

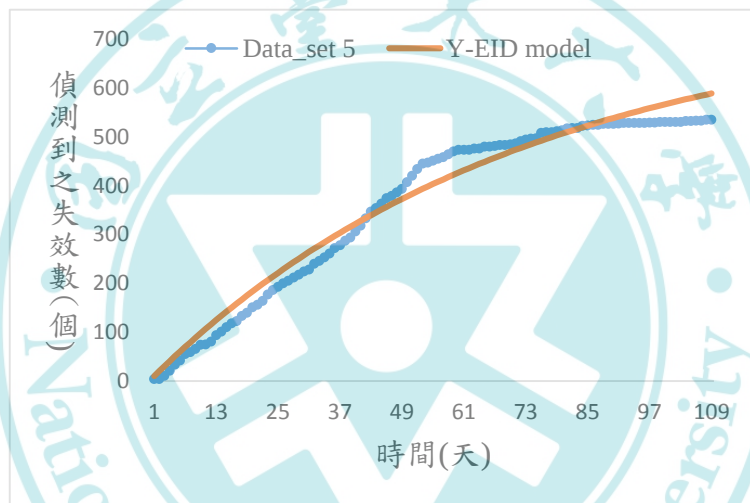


圖 62 資料集五與 Y-EID Model 之配適度曲線

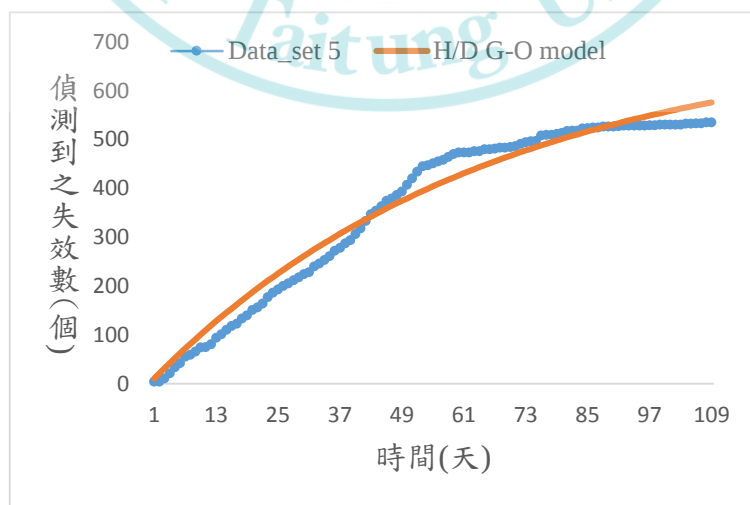


圖 63 資料集五與 H/D G-O Model 之配適度曲線

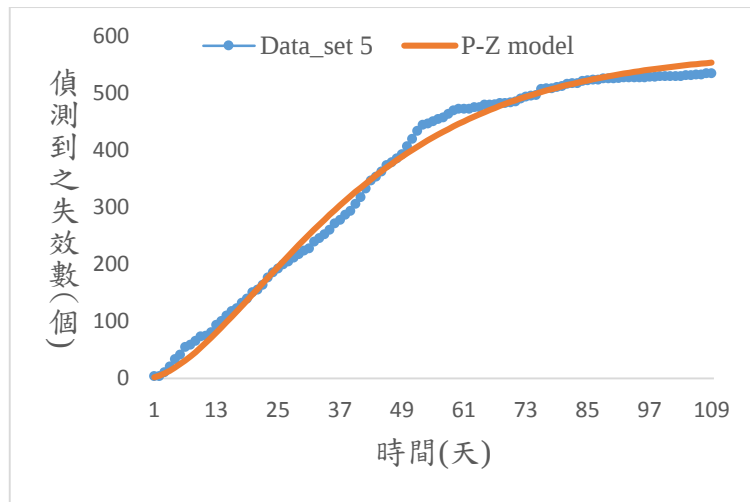


圖 64 資料集五與 P-Z Model 之配適度曲線

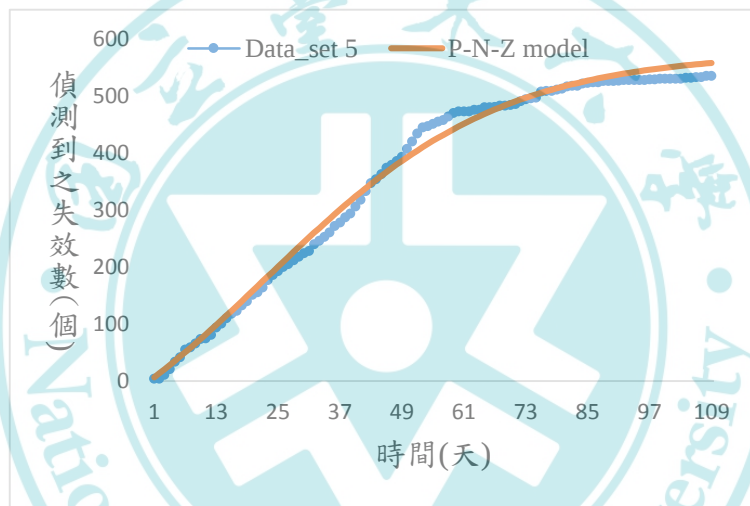


圖 65 資料集五與 P-N-Z Model 之配適度曲線

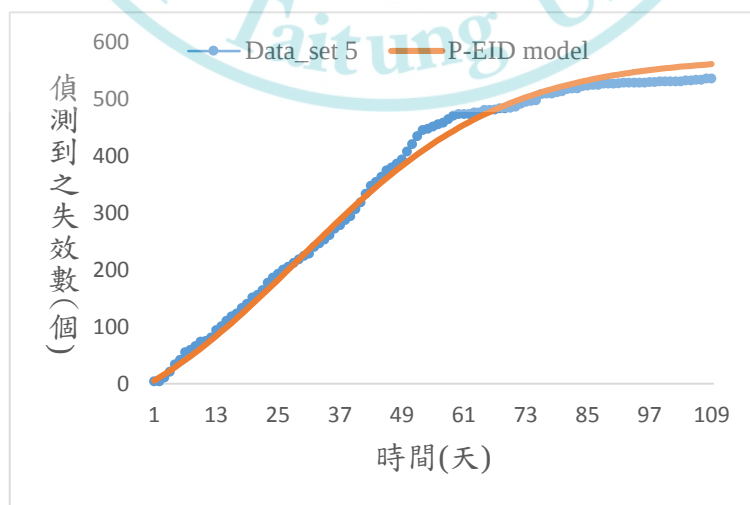


圖 66 資料集五與 P-EID Model 之配適度曲線

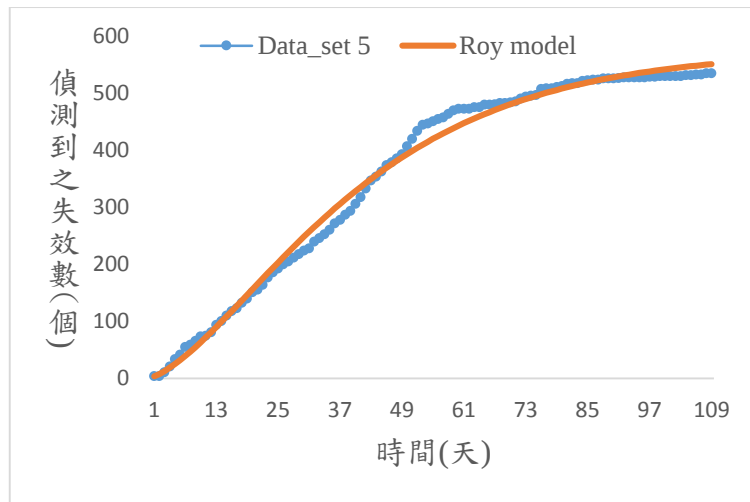


圖 67 資料集五與 Roy Model 之配適度曲線

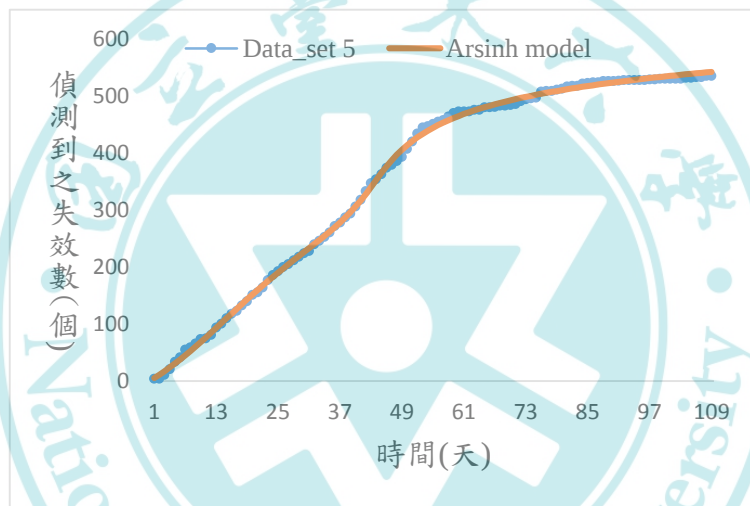


圖 68 資料集五與 Arsinh Model 之配適度曲線

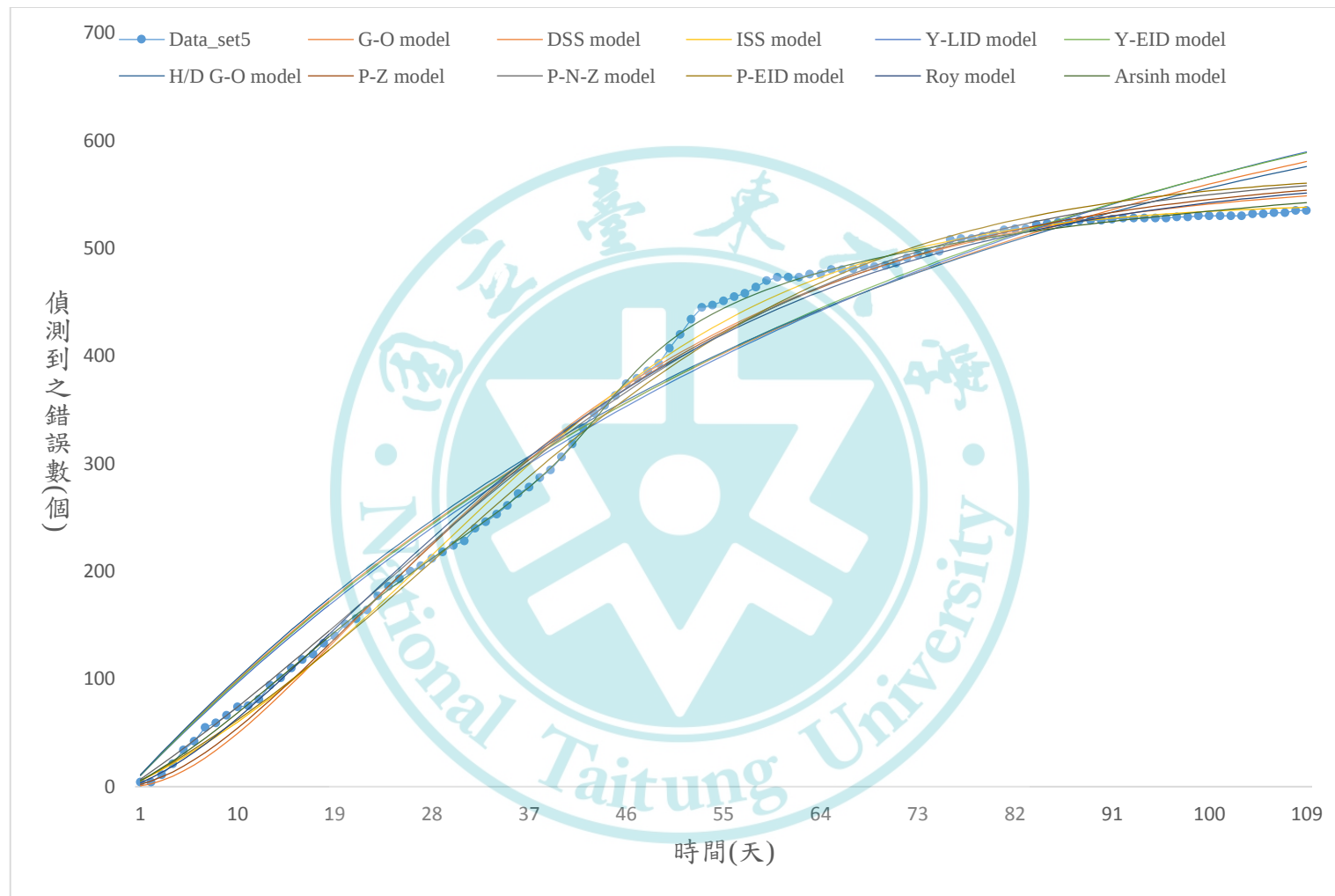


圖 69 資料集五與各模型之配適度曲線比較

表 24 各模型於資料集五的比較結果

模型\評估標準	MSE	R^2	SAE	PRV	RMSPE
Arsinh model	24.8776	0.9992*	434.9018*	4.8709*	4.8828*
G-O model	844.2060	0.9726	2805.1470	28.9204	29.4637
DSS model	226.8948	0.9926	1269.7684	14.9931	15.0256
ISS model	101.2844	0.9967	831.3997	9.9704	9.9784
Y-LID model	891.7860	0.9713	2835.2515	29.5850	30.1117
Y-EID model	886.2884	0.9715	2831.3854	29.4937	30.4270
H/D G-O model	849.2755	0.9727	2805.0689	28.8712	29.4740
P-Z model	230.6608	0.9927	1272.5532	14.9036	14.9037
P-N-Z model	235.5199	0.9925	1326.7351	15.1320	15.7245
P-EID model	236.3625	0.9925	1392.0515	15.1591	15.1719
Roy model	247.1717	0.9921	1313.2978	15.5018	15.5155

資料來源：本研究整理

各模型於資料集五的模型評估標準數值如表 24 所示。由表 24 可得知，本研究所提之模型在資料集五中，五項評估指標皆優於另外十個模型。

六、資料集六

表 25 為藉由資料集六(Data_set 6)估算出的各個現有模型與本研究所提模型之參數數值，分別帶入各自的均值函數中，即可得出各模型於各個時間點的預測累積失效數量。圖 70 至圖 80 為各模型與資料集六的配適度曲線，而圖 81 則是各模型的配適度曲線比較。

表 25 資料集六之各模型參數估計數值

模型\參數	a	b	c	α	β	γ
*Arsinh model	214.1252	0.3371	32.4752	0.6742	67.1776	-26.5778
G-O model	538.1022	0.0257	—	—	—	—
DSS model	488.1437	0.0663	—	—	—	—
ISS model	484.5938	0.0668	—	—	3.6476	—
Y-LID model	575.4006	0.0219	—	0.0001	—	—
Y-EID model	540.025	0.0255	—	0.0001	—	—
H/D G-O model	538.1036	0.0257	0.5395	—	—	—
P-Z model	485.6607	0.0639	0.003	0.8618	0.8618	—
P-N-Z model	494.4492	0.0551	—	0.0001	0.0001	—
P-EID model	—	0.0547	2.1887	493.5845	493.5845	—
Roy model	457.2334	0.06351	—	1.0727	1.0727	—

資料來源：本研究整理

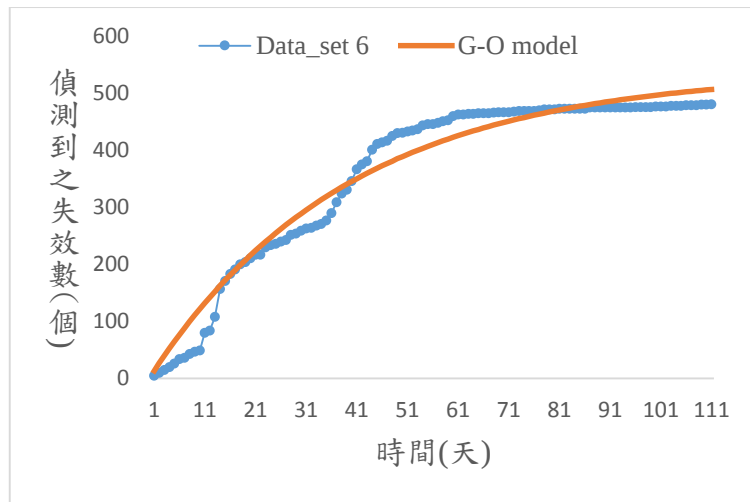


圖 70 資料集六與 G-O Model 之配適度曲線

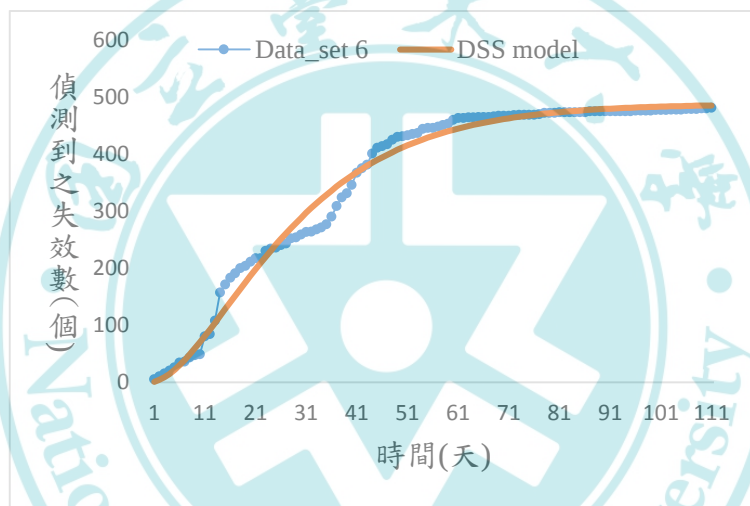


圖 71 資料集六與 DSS Model 之配適度曲線

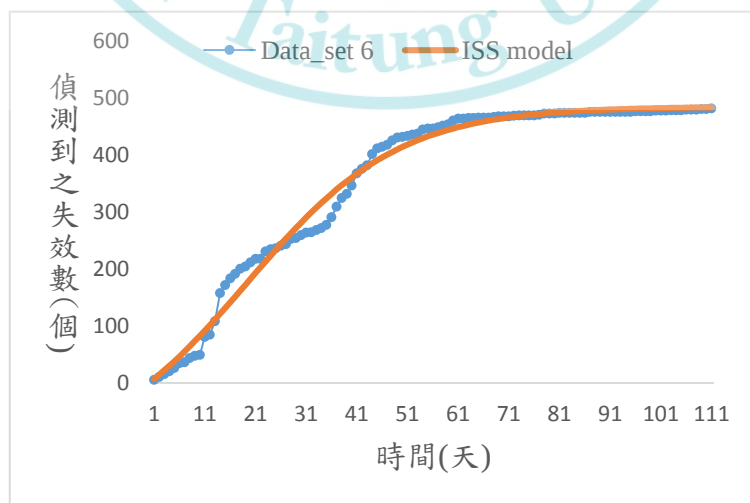


圖 72 資料集六與 ISS Model 之配適度曲線

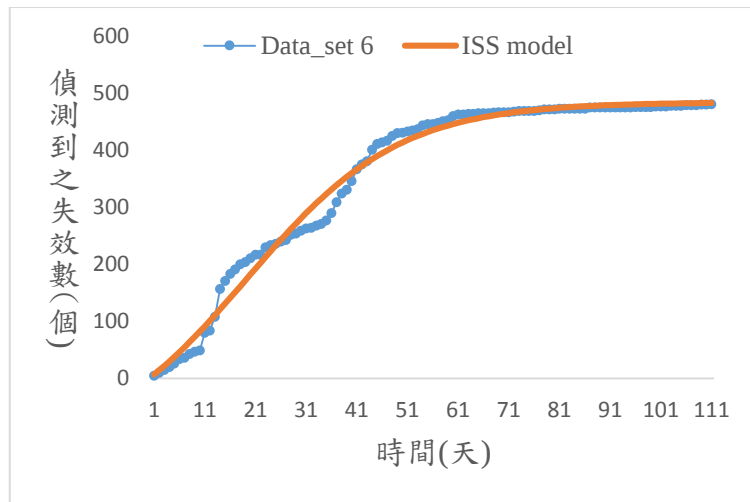


圖 73 資料集六與 Y-LID Model 之配適度曲線

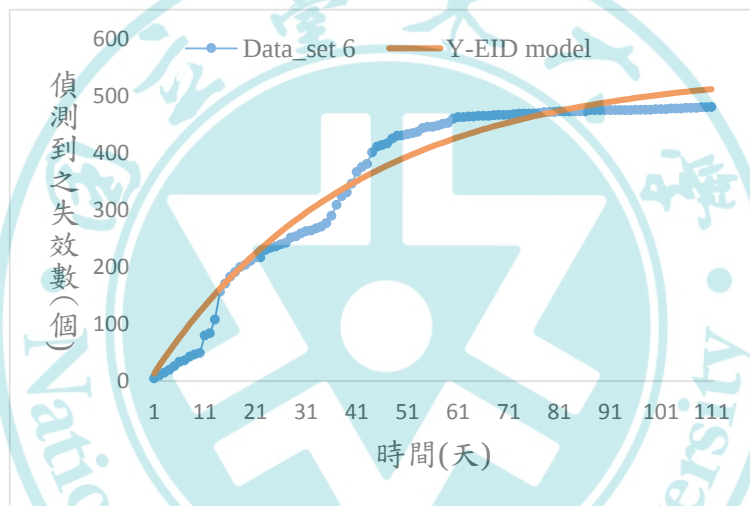


圖 74 資料集六與 Y-EID Model 之配適度曲線

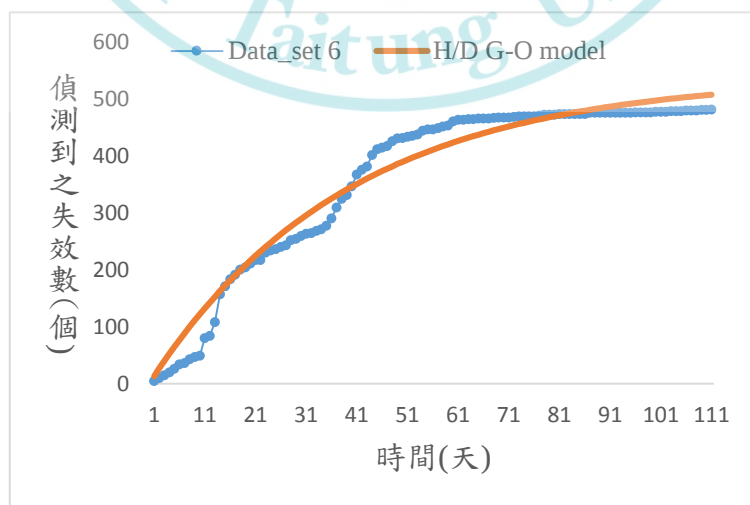


圖 75 資料集六與 H/D G-O Model 之配適度曲線

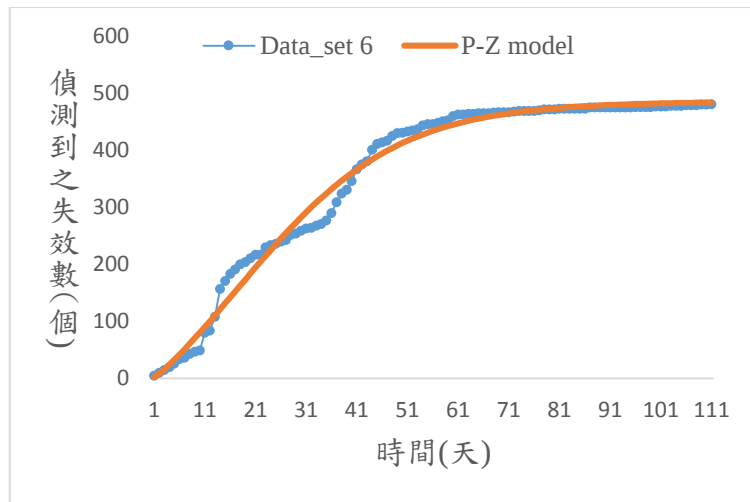


圖 76 資料集六與 P-Z Model 之配適度曲線

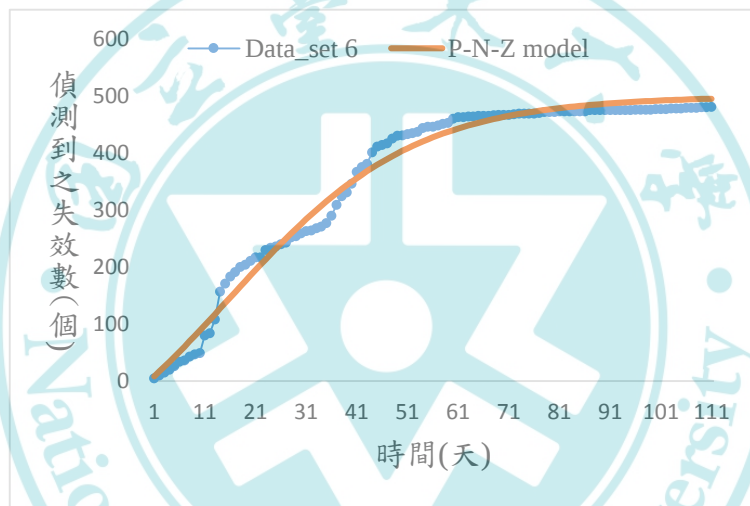


圖 77 資料集六與 P-N-Z Model 之配適度曲線

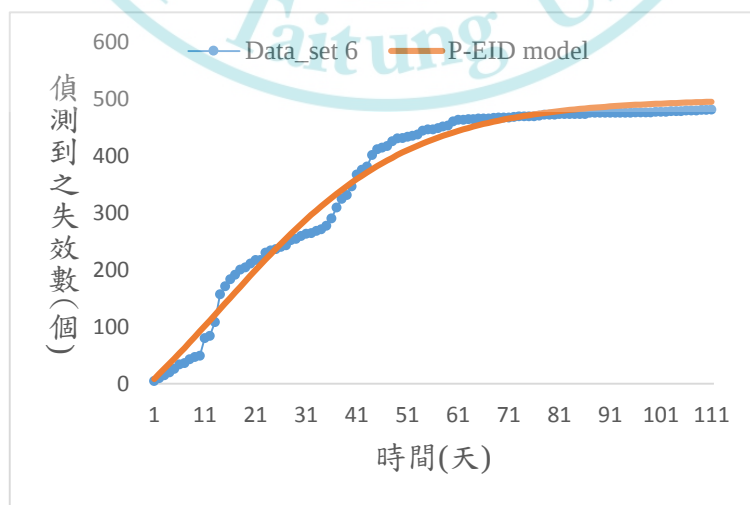


圖 78 資料集六與 P-EID Model 之配適度曲線

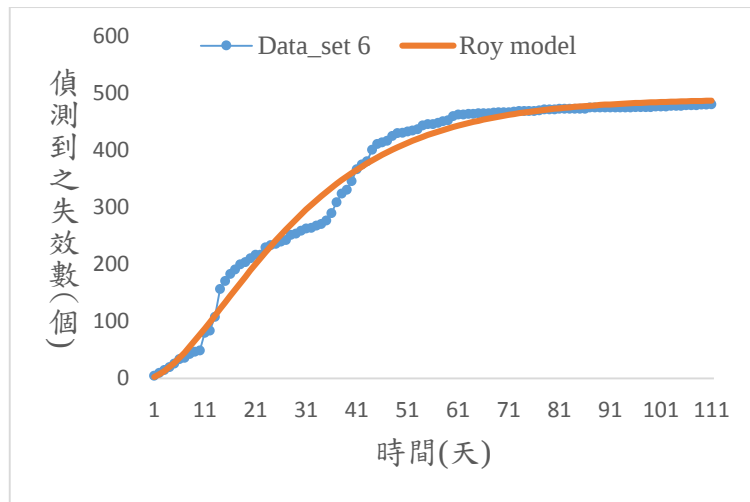


圖 79 資料集六與 Roy Model 之配適度曲線

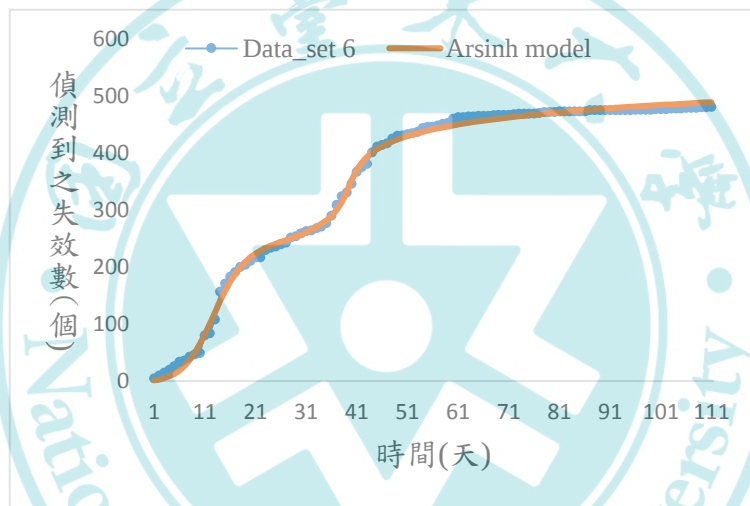


圖 80 資料集六與 Arsinh Model 之配適度曲線

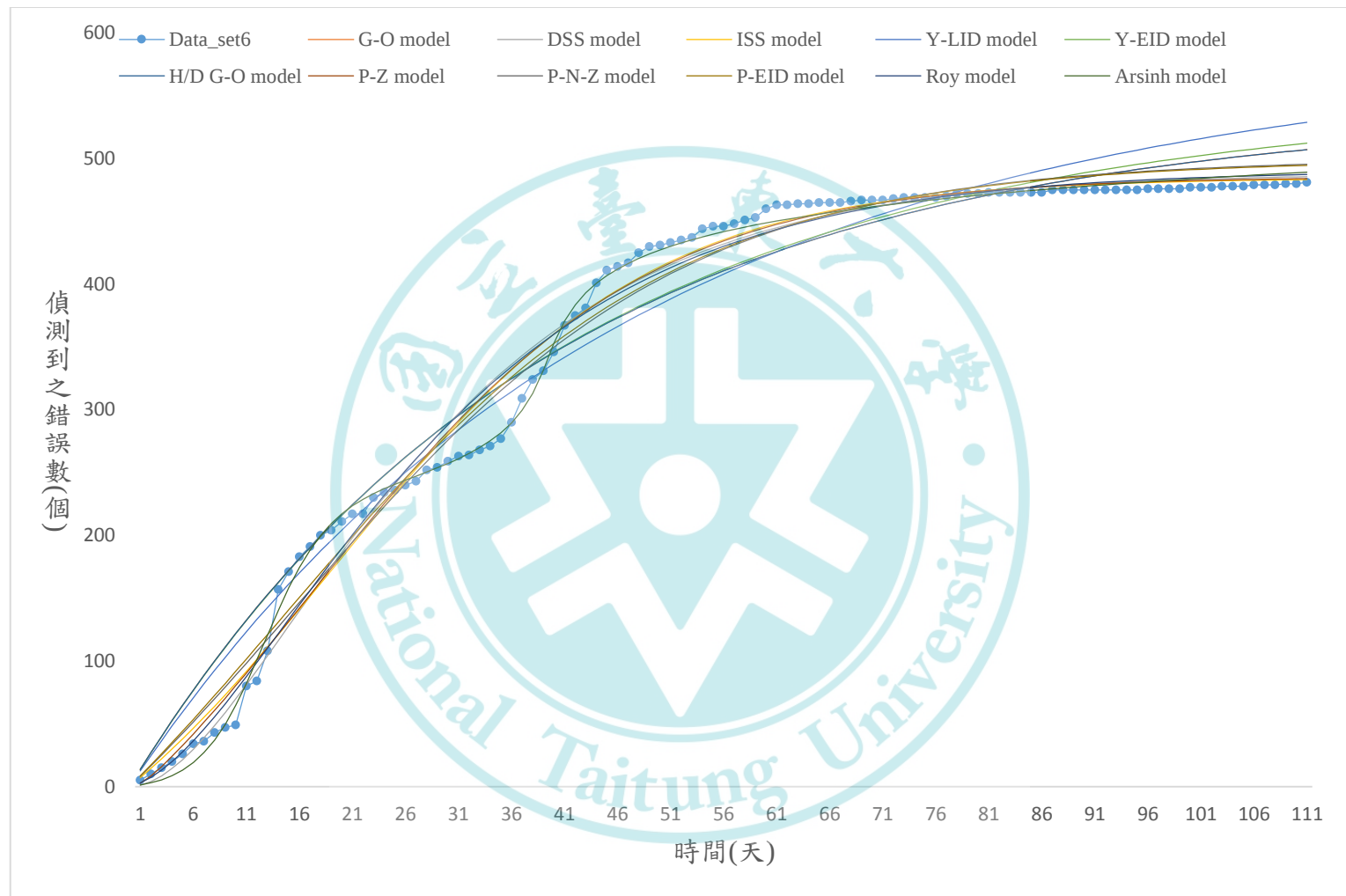


圖 81 資料集六與各模型之配適度曲線比較

表 26 各模型於資料集六的比較結果

模型\評估標準	MSE	R^2	SAE	PRV	RMSPE
Arsinh model	50.4488	0.9979*	632.5360*	6.9326*	6.9394*
G-O model	806.0257	0.9645	2604.8232	27.9770	28.2587
DSS model	329.7950	0.9855	1438.4582	18.0680	18.0774
ISS model	296.9680	0.9871	1365.6385	17.0715	17.0753
Y-LID model	957.0771	0.9583	2896.9815	30.2735	30.6507
Y-EID model	826.5578	0.9640	2655.5515	27.9003	28.4821
H/D G-O model	813.4885	0.9645	2604.8310	27.9768	28.2587
P-Z model	300.8005	0.9871	1364.0031	17.0251	17.0253
P-N-Z model	365.2281	0.9842	1793.7500	18.9200	18.9431
P-EID model	359.2478	0.9845	1769.5444	18.6485	18.7880
Roy model	330.1125	0.9857	1490.7913	17.9193	17.9195

資料來源：本研究整理

各模型於資料集六的模型評估標準數值如表 26 所示。由表 26 可得知，本研究所提之模型在資料集六中，五項評估指標皆優於另外十個模型。

七、資料集七

表 27 為藉由資料集七(Data_set 7)估算出的各個現有模型與本研究所提模型之參數數值，分別帶入各自的均值函數中，即可得出各模型於各個時間點的預測累積失效數量。圖 82 至圖 92 為各模型與資料集七的配適度曲線，而圖 93 則是各模型的配適度曲線比較。

表 27 資料集七之各模型參數估計數值

模型\參數	a	b	c	α	β	γ
*Arsinh model	21.1968	0.0923	5.0567	0.1845	258.4779	-25.4851
G-O model	580.6433	0.0005	—	—	—	—
DSS model	104.76	0.0097	—	—	—	—
ISS model	65.2752	0.0243	—	—	14.7154	—
Y-LID model	0.6493	0.0448	—	0.5194	—	—
Y-EID model	77.9908	0.0028	—	0.0054	—	—
H/D G-O model	702.2998	0.0004	0.7998	—	—	—
P-Z model	137.6896	0.0067	0.0015	0.0415	2.0822	—
P-N-Z model	0.88	0.0576	—	0.3732	9.1846	—
P-EID model	—	0.0247	2.1887	64.7023	0.0001	—
Roy model	72.6735	0.0136	—	1.0051	0.0135	—

資料來源：本研究整理

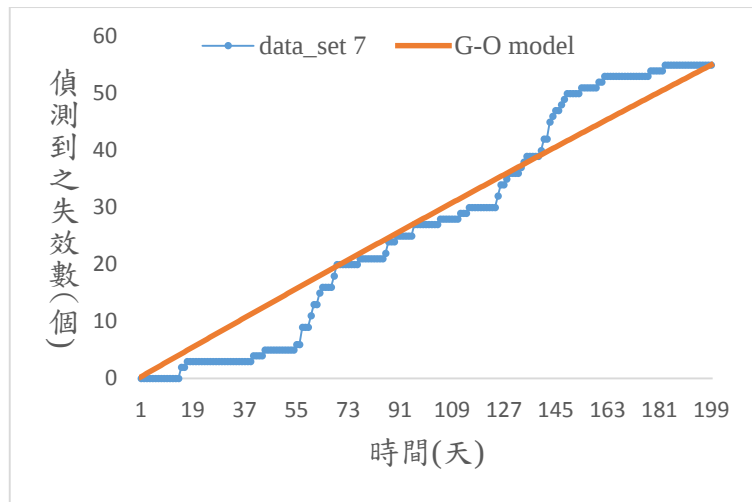


圖 82 資料集七與 G-O Model 之配適度曲線

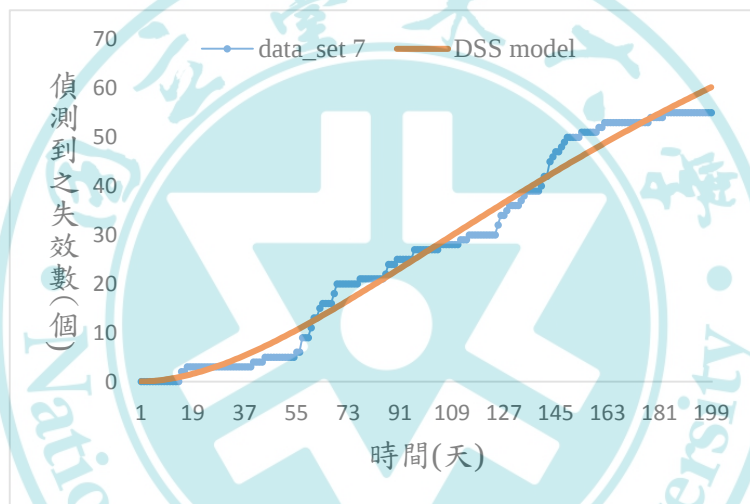


圖 83 資料集七與 DSS Model 之配適度曲線

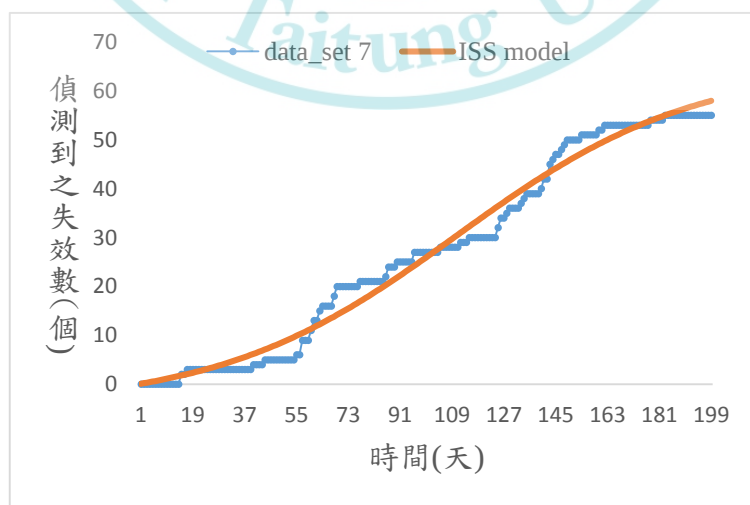


圖 84 資料集七與 ISS Model 之配適度曲線

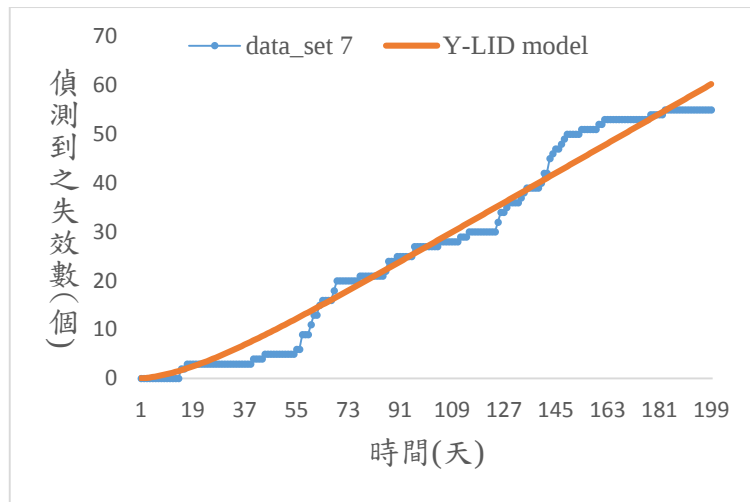


圖 85 資料集七與 Y-LID Model 之配適度曲線

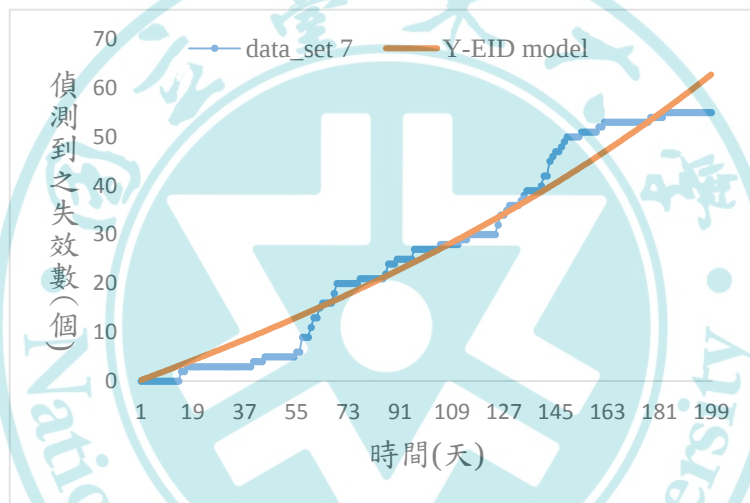


圖 86 資料集七與 Y-EID Model 之配適度曲線

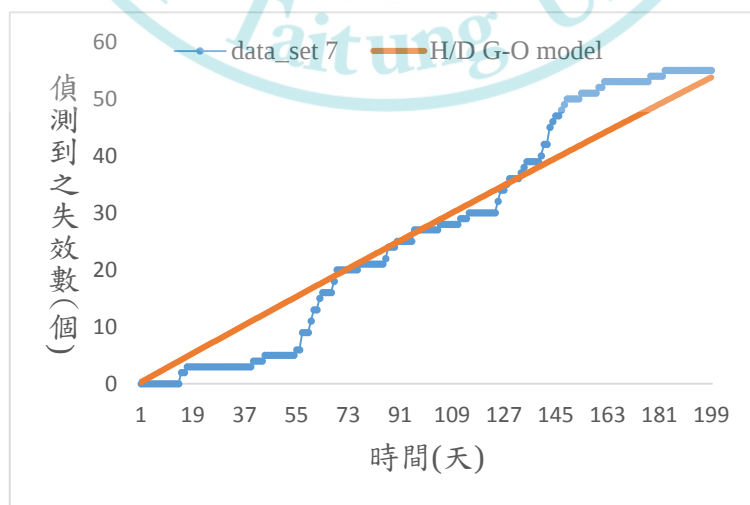


圖 87 資料集七與 H/D G-O Model 之配適度曲線

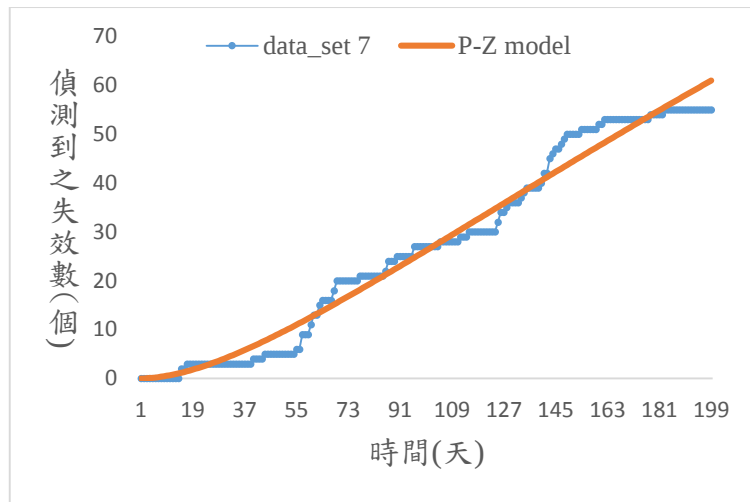


圖 88 資料集七與 P-Z Model 之配適度曲線

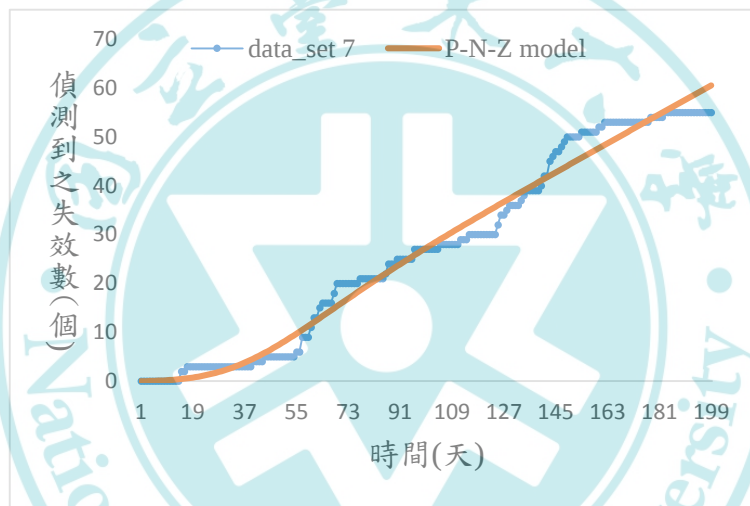


圖 89 資料集七與 P-N-Z Model 之配適度曲線

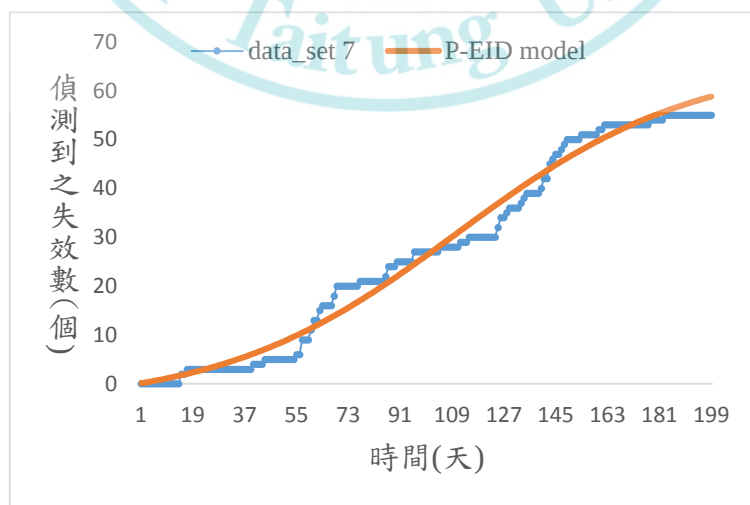


圖 90 資料集七與 P-EID Model 之配適度曲線

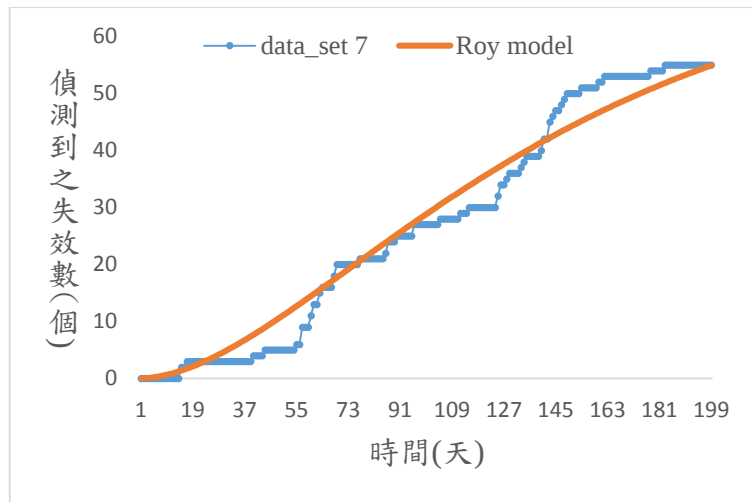


圖 91 資料集七與 Roy Model 之配適度曲線

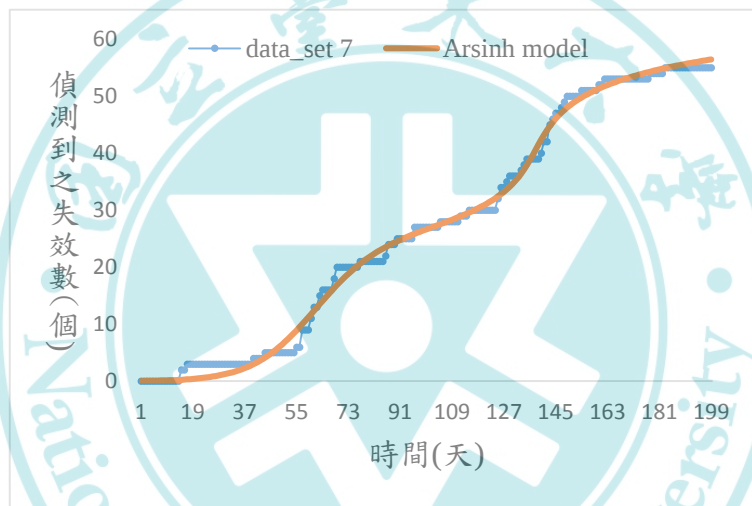


圖 92 資料集七與 Arsinh Model 之配適度曲線

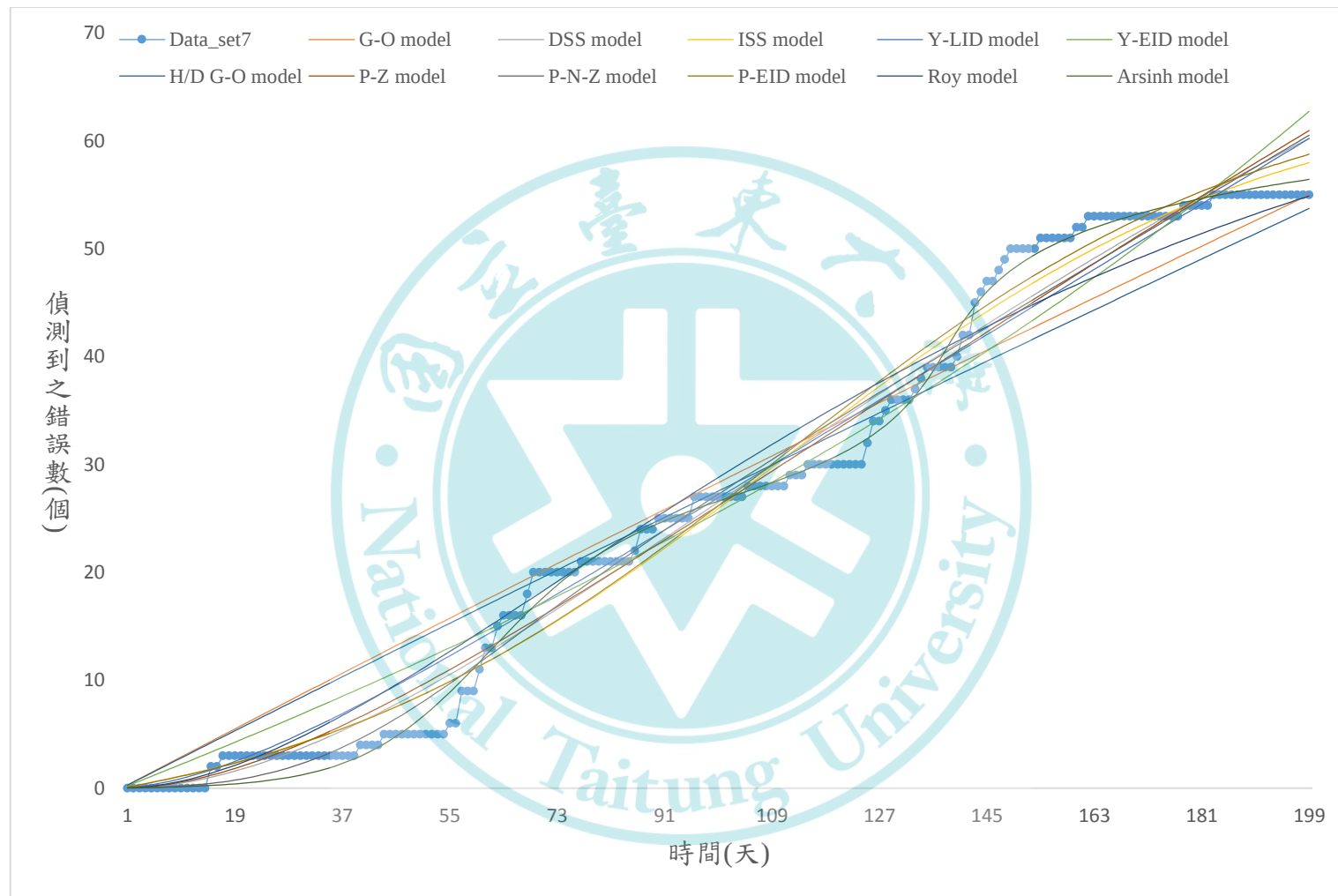


圖 93 資料集七與各模型之配適度曲線比較

表 28 各模型於資料集七的比較結果

模型\評估標準	MSE	R^2	SAE	PRV	RMSPE
Arsinh model	1.5852	0.9960*	189.1278*	1.2430*	1.2507*
G-O model	22.7288	0.9410	771.5624	4.7554	4.8464
DSS model	6.8824	0.9821	426.0718	2.6168	2.6219
ISS model	6.5414	0.9831	426.5769	2.5447	2.5489
Y-LID model	9.0854	0.9765	460.6111	2.9989	3.0317
Y-EID model	13.9115	0.9641	580.1101	3.7109	3.7309
H/D G-O model	24.5566	0.9366	783.5462	4.9304	4.9346
P-Z model	7.8521	0.9799	440.4152	2.7737	2.7751
P-N-Z model	6.9025	0.9823	412.5678	2.6073	2.6078
P-EID model	6.8052	0.9825	438.1826	2.5888	2.6285
Roy model	11.3403	0.9709	540.0378	3.3419	3.3840

資料來源：本研究整理

各模型於資料集七的模型評估標準數值如表 28 所示。由表 28 可得知，本研究所提之模型在資料集七中，五項評估指標皆優於另外十個模型。

第四節 小結

表 29 本研究所提模型於各組數據之評估標準綜合表現排名表

資料集\評估標準	MSE	R^2	SAE	PRV	RMSPE
資料集一	2	1	1	1	1
資料集二	1	1	1	1	1
資料集三	1	1	1	1	1
資料集四	1	1	1	1	1
資料集五	1	1	1	1	1
資料集六	1	1	1	1	1
資料集七	1	1	1	1	1

資料來源：本研究整理

透過以上分析與比較，可發現本研究所提之模型於七組實際資料的測試中，皆得到良好的配適度曲線，表示研究所提之模型能夠預測軟體的實際失效行為；而在評估標準方面，本研究所提之模型與另外十個現有模型的比較，除了資料集一的 MSE 指標表現不如 ISS Model 之外，其餘四項指標於七組測試資料中皆表現得優於另外十個模型，而在資料集二至資料集七剩餘六組數據集之中，本研究所提之模型的 MSE 指標也都是優於各個模型的，因此，本研究推論所提之模型於大部分情況下擁有比另外十個模型更加準確的預測能力。

而本研究挑選的前四組實際資料特意選擇了的時間間隔分別為小時、天、週和月的資料，用以驗證本研究所提之模型能夠穩定的於各種不同的時間間隔內進行有效的預測；而後三組實際資料則為記錄數量較多筆的失效資料(皆超過 100 筆數據)，以驗證本研究所提之模型能夠準確的在較大尺度的資料量下進行有效的預測；經過資料的分析以及模型的比較，證實本研究所提之模型在面對長時間與大尺度實際數據時，仍保有良好的預測能力；因此，就整體而言，本研究認為所提之模型能夠於大部份情境下有效的預測軟體失效的發生。

其中特別值得注意的是第六組與第七組失效數據的分析結果，此二組資料具有兩次的上升趨勢變化，而本研究所提之模型仍能預測其趨勢，證明該模型有能力預測趨勢變化較為複雜的實際資料。

而關於資料集一中，MSE 指標的表現略遜於 ISS Model 的原因，本研究認為其主因是因為參數使用的數量，本研究所提之模型參數數量有六項，而 ISS Model 只有三項，但由 SAE 可得知，本研究所提之模型預估的絕對誤差並沒有明顯小於 ISS Model，導致 MSE 指標於資料集一中判定 ISS Model 優於本研究所提之模型；此情況非常值得反思，建構模型之初，為期望模型更加符合真實情況而加入許多參數，但反而造成模型不易操控，而且在某些情況下的表現並不如預期。雖然本研究所提之模型於其他指標的表現仍優於 ISS Model，但若是遇到較為簡單的情境，如：「接近完美除錯」…等，或許選擇 ISS Model 可迅速解決問題。

第五章 結論與建議

在這軟體廣泛應用的時代，軟體可靠度受到的重視格外突出，時至今日，軟體可靠度仍是學界致力研究的重要項目；過去數十年，許多學者為軟體可靠度領域的研究做出卓越的貢獻，而本研究藉由以往學者們所提出的理論基礎，研發並建構新的軟體可靠度成長模型，期望藉此提高軟體的可靠度，並對學術做出貢獻。透過本研究的分析與比較，總結歸納以下結論與建議。

第一節 結論

本研究以非齊次卜瓦松過程做為基礎，建立了一個考慮錯誤生成、除錯對於錯誤生成的抑制效果、錯誤偵測的學習效果與反雙曲正弦函數特性的不完美除錯軟體可靠度成長模型。並嘗試建構除錯對於錯誤生成有所抑制效果的表示方法模型；與對於錯誤偵測的學習效果分析及賦予其學習效果的方法。透過七組真實失效數據集進行驗證，並與十個過去學者所提出的模型進行比較與分析，結果顯示，本研究所提之模型對實際的失效情況具有良好的預測能力，並且於大部分情況下，本研究所提之模型的預測能力優於比較的十個模型。因此，本研究推論，所提之模型具備有效預測失效的能力，也確認，所提的兩個特性能夠有效的提升模型的預測能力，故本研究認為考慮學習效果、除錯對於錯誤生成的抑制效果以及良好的趨勢函數能夠有效的建立出軟體可靠度成長模型。

第二節 研究限制

本研究考量研究時間、使用方法以及數據…等各類因素後，進行彙整及分析，並列出以下幾點研究限制：

1. 本研究對於不完美除錯的假設，僅考慮錯誤生成；因此，並未深入探討錯誤的排除與錯誤產生之間的關係。
2. 本研究對於除錯與錯誤生成之間的關係僅定義為除錯將抑制錯誤生成，也許除錯與錯誤生成之間的關係存在其他運作模式。
3. 本研究中，函數導入方法為本研究根據基本公式進行推論與假設後所提出，或許除此方法之外，仍存在其他方法可以使得模型具有豐富的函數

特性。

4. 本研究對於錯誤偵測率的學習效果假設為 S 型遞增，可能存在其他模式能夠描述學習效果。
5. 本研究基於測試時間，測試範圍僅以七組實際資料進行比較；因此，在其他資料上的預測狀況不見得每次都能優於其他模型。
6. 本研究基於測試時間，測試結果僅與十組較為常見的現有基本模型進行比較；因此，不能保證每次預測的結果皆優於現行存在之所有模型。

第三節 建議

藉由分析結論與研究限制，本研究提出以下幾點建議，給予未來研究做為參考：

1. 由於本研究在不完美除錯的考慮中僅考量錯誤生成，但未考慮錯誤的排除；因此，對於未來研究中，或許能夠加入考慮不完美故障排除或是排除延遲的情況。
2. 關於錯誤產生率方面，許多研究將錯誤產生率定義為常數，在未來研究中，或許可以考慮將錯誤產生率定義為一個隨時間而產生變化的函數，或許能更有效的描述錯誤的生成過程。
3. 關於除錯與錯誤生成的相互關係，本研究僅假設除錯對於錯誤生成有所抑制；或許未來可以更深入的研究錯誤的排除以及學習效果對於錯誤生成的影響，並對其進行詳細的分析。
4. 關於函數選擇，本研究選擇導入不收斂的反雙曲正弦函數，起初是考量了現實中，錯誤難以完全排除，以及期望可以描述模糊學習(Fuzzy Learning)效果等因素，但最終模型並未能夠明顯區分所考慮的因素造成的影響。而未被選中的函數中，也許選取反正切函數(Inverse Tangent Function, $\arctan$)或是某些鐘形(Bell Shaped)函數的積分如：累積分布函數(Cumulative Distribution Function)……等，做為導入的函數，可能會有不錯的預測效果。

5. 關於錯誤偵測的學習效果，本研究賦予學習效果僅嘗試了初始偵測率為常數的情況，並做為限制條件，但事實上還有許多函數帶入賦予學習效果的公式後，同樣具備 S 形遞增的特性，因此，於未來研究中，或許能夠尋找更能完整描述學習效果的函數，來提升模型的預測力。
6. 若時間允許，或許能夠進行更多組的失效數據測試，與更多不同模型進行比較，以準確評估模型的預測能力。



參考文獻

中文部分

書籍：

彭鴻霖 (2000)。可靠度技術手冊。中華民國品質學會，臺北市。

博士論文：

邱桂珍 (2008)。學習效果觀點之軟體可靠度成長模型。博士論文，國立成功大學工業與資訊管理研究所，台南市。

劉建邦 (2019)。可靠度研究及失效分析於微電子封裝中的銅-鋁鍵合系統。博士論文，國立成功大學微電子工程研究所，台南市。

碩士論文：

王育浚 (2017)。符合 ISO 26262 功能安全需求之機電系統可靠度及其提升方法——以馬達驅動器為例。碩士論文，國立臺灣大學機械工程學研究所，台北市。

王亭雅 (2015)。具指數型、線性與 S 型特性之不完美除錯軟體可靠度模型。碩士論文，國立臺東大學資訊管理學系碩士班，台東縣。

王若君 (2018)。基於條件均值譜之跨河橋樑可靠度研究。碩士論文，國立臺灣科技大學營建工程系，台北市。

江文馨 (2013)。具指數型及線性特性之不完美除錯軟體可靠度模型。碩士論文，國立臺東大學資訊管理學系碩士班，台東縣。

李嘉華 (2014)。具有除錯延遲與測試心力函數之不完美除錯軟體可靠度模型。碩士論文，國立臺東大學資訊管理學系碩士班，台東縣。

林邑築 (2018)。多種相關失效不完美除錯模型下軟體推出時間。碩士論文，國立成功大學工業與資訊管理學系，台南市。

- 林哲宇 (2019)。考慮不完美除錯與錯誤生成的軟體可靠度成長模型。碩士論文，國立臺東大學資訊管理學系碩士班，台東縣。
- 周峰成 (2001)。1/f 型態隨機點過程之應用-以台灣地震記錄為例。碩士論文，東海大學統計學系，台中市。
- 張琇惟 (2008)。團隊動作互動中學習曲線效果之實驗研究。國立中正大學企業管理研究所，嘉義縣。
- 莊滿如 (2018)。考量時間限制下之多階狀態多產品物流網路可靠度。碩士論文，國立臺灣科技大學工業管理系，台北市。
- 葉芳瑜 (2019)。模擬 1/f 神經傳導流及集群卜瓦松過程之參數估計-以正常人心跳為例。碩士論文，東海大學統計學系，台中市。
- 賈奕騰 (2014)。跨河橋梁耐洪能力之可靠度分析。碩士論文，國立臺灣科技大學營建工程系，台北市。
- 蔡政儒 (2015)。結合硬體修復與錯誤修正碼提升嵌入式記憶體良率與可靠度技術。碩士論文，國立臺灣科技大學電機工程系，台北市。
- 謝協成(2020)。使用適應性容錯技術以提升電阻式記憶體之良率及可靠度。碩士論文，國立臺灣科技大學電機工程系，台北市。
- 謝濱壕 (2019)。具三種故障內容函數特性並考慮不完美除錯之軟體可靠度成長模型。碩士論文，國立臺東大學資訊管理學系碩士班，台東縣。
- 顧家齊 (2013)。可提升影像處理電路良率與可靠度之方法及其硬體設計。碩士論文，國立中山大學電機工程學系研究所，高雄市。

外文部分

Books:

- Kapur, P. K., Pham, H., Gupta, A. & Jha, P.C. (2011b). Software Reliability Assessment with OR Applications. Springer Series in Reliability Engineering. Springer-Verlag London Limited.
- Lyu, M. R. (1996). Handbook of Software Reliability Engineering. IEEE Computer Society Press. <http://www.cse.cuhk.edu.hk/~lyu/book/reliability/index.html>
- Musa, J.D., Iannino, A., Okumoto, K. (1987). Software Reliability Measurement, Prediction, Application, Int. Ed. McGraw-Hill.
- Pham, H. (2006). System Software Reliability. Springer-Verlag London Limited.
- Poong, H. S. (2009). Reliability and Risk Issues in Large Scale Safety-critical Digital Control Systems. Springer Series in Reliability Engineering. Springer-Verlag London Limited.
- Taylor, H. M. & Karlin, S. (1998). An Introduction to Stochastic Modeling Third Edition. San Diego: Academic Press.
- Watkins, J. C. (2016). An Introduction to the Science of Statistics: From Theory to Implementation. Preliminary Edition, Tucson, AZ: University of Arizona, 1-429. <https://www.math.arizona.edu/~jwatkins/statbook.pdf>

Journal Articles:

- Aggarwal, A. G., Kapur, P. K. & Garmabaki, A. S. (2011). Imperfect Debugging Software Reliability Growth Model for Multiple Releases. Proceedings of the 5th National Conference on Computing for Nation Development—INDIAcom, pp.337-344.
- Alhazzaa, L. & Andrews, A. A. (2019). A Systematic Mapping Study on Software Reliability Growth Models that Consider Evolution. Int'l Conf. Software Eng. Research and Practice | SERP'19 |, pp.83-90.

- Chatterjee, S. & Shukla, A. Effect of Test Coverage and Change Point on Software Reliability Growth Based on Time Variable Fault Detection Probability. *Journal of Software*, 11(1), pp.110-117.
- Chiu, K. C. (2013). A discussion of software reliability growth models with time-varying learning effects. *American Journal of Software Engineering and Applications*, 2(3), pp.92-104.
- Chiu, K. C., Huang, Y. S. & Lee, T. Z. (2008). A study of software reliability growth from the perspective of learning effects. *Reliability Engineering and System Safety*, 93, pp.1410-1421.
- Chopra, S., Nautiyal, L., Malik, P., Ram, M. & Sharma, M. K. (2020). A Non-Parametric Approach for Survival Analysis of ComponentBased Software. *International Journal of Mathematical, Engineering and Management Sciences*, 5(2), pp.309-318.
- Gandhi, N., Gondwal, N. & Tandon, A. (2018). Reliability Modeling of OSS Systems based on Innovation-Diffusion Theory and Imperfect Debugging. *Proceedings of the First International Conference on Information Technology and Knowledge Management*, 14, pp.53-58.
- Goel, A. L. & Okumoto, K. (1979). Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures. *IEEE Transactions on Reliability*, 28(3), pp.206-211.
- Grottke, M. (2001). Software Reliability Model Study. IST-1999-55017, pp.1-41.
- Hanagal, D. D. & Bhalerao, N. N. (2018). Analysis of Delayed S Shaped Software Reliability Growth Model with Time Dependent Fault Content Rate Function. *Journal of Data Science*, 16(4), pp.857-878.
- Hanagal, D. D. & Bhalerao, N. N. (2019). Modeling on Generalized Extended Inverse Weibull Software Reliability Growth Model. *Journal of Data Science*, 17(3), pp.575-592.
- Hossain, S. A. & Dahiya, R. C. (1993). Estimating the Parameters of a Non-homogeneous Poisson-Process Model for Software Reliability. *IEEE Transactions on Reliability*, 42(4), pp.604-612.

- Jeske, D. R. & Zhang, X. (2005). Some successful approaches to software reliability modeling in industry. *The Journal of Systems and Software*, 74, pp.85-99.
- Joshi, S. L., Deshpande, B. & Punnekkat, S. (2019). Experimental Analysis of Dependency Factors of Software Product Reliability using SonarQube. *International Workshop on Software Measurement and International Conference on Software Process and Product Measurement 2019*, 2476, pp.130-137.
- Kapur, P. K., Goswami, D.N., Bardhan, A. & Singh, O. (2008). Flexible software reliability growth model with testing effort dependent learning process. *Applied Mathematical Modelling*, 32, pp.1298-1307.
- Kapur, P. K., Pham, H., Anand, S., Yadav, K. (2011a). A Unified Approach for Developing Software Reliability Growth Models in the Presence of Imperfect Debugging and Error Generation. *IEEE Transactions on Reliability*, 60(1), pp.331-340.
- Lee, D. H., Chang, I. H., Pham, H. & Song, K. Y. (2018). A Software Reliability Model Considering the Syntax Error in Uncertainty Environment, Optimal Release Time, and Sensitivity Analysis. *Appl. Sci.*, 8(9), 1483, pp.1-17.
- Ohba, M. (1984a). Inflection S-shaped Software Reliability Growth Model. *Stochastic Models in Reliability Theory*, Springer-Verlag Berlin Heidelberg, pp.144-162.
- Ohba, M. (1984b). Software reliability analysis models. *IBM J. RES. DEVELOP*, 28, pp.428-443.
- Park, J. & Baik, J. (2015). Improving software reliability prediction through multi-criteria based dynamic model selection and combination. *The Journal of Systems and Software*, 101, pp.236-244.
- Pham, H. (2016). A generalized fault-detection software reliability model subject to random operating environments. *Vietnam J Comput Sci*, 3, pp.145–150.

- Pham, H., Nordmann, L. & Zhang, X. (1999). A General Imperfect-Software-Debugging Model with S-Shaped Fault-Detection Rate. *IEEE Transactions on Reliability*, 48(2), pp.169-175.
- Pham, H. & Zhang, X. (1997). An NHPP Software Reliability Model and Its Comparison. *International Journal of Reliability, Quality and Safety Engineering*, 4(3), pp.269-282.
- Rafi, S. M. & Akthar, S. (2011). Software Reliability Growth Model with Bass Diffusion Test-Effort Function and Analysis of Software Release Policy. *International Journal of Computer Theory and Engineering*, 3(5), pp.671-680.
- Rao, K. M. & Anuradha, K. (2017). An Efficient Method for Enhancing Reliability and Selection of Software Reliability Growth Model through Optimization Techniques. *Journal of Software*, 12(1), pp.1-18.
- Roy, P., Mahapatra, G. S. & Dey, K. N. (2014). An NHPP Software Reliability Growth Model with Imperfect Debugging and Error Generation. *International Journal of Reliability, Quality and Safety Engineering*, 21(2), pp.1-32.
- Sahu, K. & Srivastava, R. K. (2020). Needs and Importance of Reliability Prediction: An Industrial Perspective. *Information Sciences Letters*, 9(1), pp.33-37. www.naturalspublishing.com/Journals.asp
- Sun, B., Li, Y., Wang, Z., Li, Z., Xia, Q., Ren, Y., Feng, Q., Yang, D. & Qian, C. (2020). Physics-of-failure and computer-aided simulation fusion approach with a software system for electronics reliability analysis. *Eksploracja i Niezawodnosc – Maintenance and Reliability* 2020,22(2), pp.340–351.
- Tohma, Y., Tokunaga, K., Nagase, S. & Murata, Y. (1989). Structural Approach to the Estimation of the Number of Residual Faults Based on the Hyper-Geometric Distribution. *IEEE Transactions on Software Engineering*, 15(3), pp. 345-355.

- Tohma, Y., Yamano, H., Ohba, M. & Jacoby, R. (1991). Parameter Estimation of the Hyper-Geometric Distribution Model for Real Test/Debug Data. Proceedings of the 1991 IEEE International Symposium on Software Reliability Engineering, pp. 28-34.
- Wood, A. (1996). Software Reliability Growth Models. Technical Computers, 96.1, pp. 1-30.
- Wright, T. P. (1936). Factors Affecting the Cost of Airplanes. Journal of the Aeronautical Sciences, 3, pp.122-128.
- Xu, J. & Yao, S. (2016). Software Reliability Growth model with Partial Differential Equation for Various Debugging Processes. Math. Probl. Eng., 2016, pp.1–13.
- Yamada, S., Ohba, M. & Osaki, S. (1983). S-Shaped Reliability Growth Modeling for Software Error Detection. IEEE TRANSACTIONS ON RELIABILITY, 32(5), pp.475-484.
- Yamada, S., Tokuno, K. & Osaki, S. (1992). Imperfect debugging models with fault introduction rate for software reliability assessment. INT. J. Systems SCI., 23(12), pp.2241-2252.
- Zhang, X., Teng, X. & Pham, H. (2003). Considering Fault Removal Efficiency in Software Reliability Assessment. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, 33(1), pp.114-120.

Others:

- ISO/IEC/ IEEE (2010). Systems and software engineering — Vocabulary. International Standard ISO/IEC/ IEEE 24765:2010(E). First edition.

附錄

附錄一：故障內容函數之限制條件推導過程

由假設以及導入的函數所得到故障內容函數 $a(t)$ ，此函數屬於累積遞增函數，因此必須滿足微分恆大於 0 的限制條件，此限制條件整理過程如下：

$$\frac{da(t)}{dt} \geq 0, \text{ for any } t \in \mathbb{N}$$

$$\begin{aligned} \frac{da(t)}{dt} &= \left(a + c \left[\operatorname{arsinh}(at + \gamma) + \frac{\alpha}{b\sqrt{(at + \gamma)^2 + 1}} - \operatorname{arsinh}(\gamma) - \frac{\alpha}{b\sqrt{\gamma^2 + 1}} \right] \right)' \\ &= c \left[\operatorname{arsinh}(at + \gamma) + \frac{\alpha}{b\sqrt{(at + \gamma)^2 + 1}} \right]' \\ &= c\alpha((at + \gamma)^2 + 1)^{-\frac{1}{2}} - \frac{c\alpha^2}{b}(at + \gamma)((at + \gamma)^2 + 1)^{-\frac{3}{2}} \geq 0 \\ &\rightarrow \left[c\alpha((at + \gamma)^2 + 1)^{-\frac{1}{2}} - \frac{c\alpha^2}{b}(at + \gamma)((at + \gamma)^2 + 1)^{-\frac{3}{2}} \right] \times \frac{b((at + \gamma)^2 + 1)^{\frac{3}{2}}}{c\alpha} \geq 0 \times \frac{b((at + \gamma)^2 + 1)^{\frac{3}{2}}}{c\alpha} \\ &\rightarrow b((at + \gamma)^2 + 1) - \alpha(at + \gamma) = b(at + \gamma)^2 - \alpha(at + \gamma) + b \geq 0 \end{aligned}$$

為保持 $\frac{da(t)}{dt}$ 處處大於 0，以上方程式必須為恆正，因此它的判別式必須小於等於 0，經整理後公式如下：

$$\alpha^2 - 4b^2 \leq 0$$

$$\rightarrow \alpha^2 \leq 4b^2$$

$$\rightarrow -2b \leq \alpha \leq 2b$$

由於 α 必須大於 0，因此限制條件修正為 $0 < \alpha \leq 2b$ 。