

南華大學科技學院資訊管理學系

碩士論文

Department of Information Management

College of Science and Technology

Nanhua University

Master Thesis

基於 Python 深度學習之中醫證候與氣象變化的
預測性分析

Predictive Analysis of TCM Syndromes and
Meteorological Changes Based on Python Deep Learning

石家綸

Chia-Lun Shih

指導教授：邱宏彬 博士

Advisor: Hung-Pin Chiu, Ph.D.

中華民國 110 年 7 月

July 2021

南華大學
科技學院資訊管理學系
碩士學位論文

基於Python深度學習之中醫證候與氣象變化的預測性分析
Predictive analysis of TCM syndromes and meteorological
changes based on Python deep learning

研究生：石家綸

經考試合格特此證明

口試委員：林迺謙
葉明憲
邱宏彬

指導教授：邱宏彬

系主任(所長)：陳信良

口試日期：中華民國 110 年 6 月 24 日

南華大學資訊管理學系碩士論文著作財產權同意書

立書人：石家綸之碩士畢業論文

中文題目：

基於Python深度學習之中醫證候與氣象變化的預測性分析

英文題目：

Predictive Analysis of TCM Syndromes and Meteorological Changes Based
on Python Deep Learning

指導教授： 邱宏彬 博士

學生與指導老師就本篇論文內容及資料其著作財產權歸屬如下：

- 共同享有著作權
- 共同享有著作權，學生願「拋棄」著作財產權
- 學生獨自享有著作財產權

學 生：石家綸 (請親自簽名)

指導老師：邱宏彬 (請親自簽名)

中華民國 110 年 7 月 10 日

誌 謝

感謝我的指導教授—邱宏彬老師，從大學一年級開始到碩士二年級老師教導了很多不同的資訊技術，像是程式語言、資料庫、網頁等，在大四到碩二期間擔任過老師的助教，在課堂後，輔導學生們程式語言，除了幫助他們理解，也讓我基礎更牢固。

感謝資管系的老師們在大學及碩士的課堂中，讓我獲取到很多的知識。

感謝我的碩士同學們，會提醒一些需要知道的資訊，像是助教的月報表繳交時間、助教的活動時間等。

感謝學長雖然畢業了，但在我撰寫論文的期間還是會用社群軟體鞭策我。

感謝口試委員，林迺衛老師與葉明憲醫師，因為有口委們的建議，讓我理解到論文還有不足之處，能將論文修改至更好。

最後感謝家人及朋友們不斷照顧與支持我，讓我順利地從碩士畢業。

石家綸 謹誌
民國一零九年六月

基於 Python 深度學習之中醫證候與氣象變化的 預測性分析

學生：石家綸

指導教授：邱宏彬

南華大學 資訊管理學系碩士班

摘 要

近年來，各領域在執行策略上開始顧慮氣象因素，進而有氣象經濟、氣象醫學等與氣象相關的事物出現。中醫自古以來非常重視氣象和身體變化的關係，所有的病因皆是外在因子的風、寒、暑、濕、燥、火等六氣與身體相應的結果。而中醫又以辨證論治為主要的診斷方法，每一種證候都包含了外在病因及身體內部臟腑的結合。故本研究以肝火上炎為主要目標，透過不同神經元個數、時間長度、資料型態與天氣因子數量，找出適合之深度學習模型。

研究結果顯示：(1)時間長度以過去 14 天為基準，天氣因子以平均氣壓與平均氣溫雙因子作為樣本資料，透過 4 組不同神經元個數之比較，得出神經元個數 32、16 之人數證候比例測試集之誤

差，在其他 3 組神經元個數中最小。(2)選用神經元個數 32、16 與另外時間長度 7、21 進行 DNN、RNN、LSTM 三種模型之比較，得出在神經元個數 32、16 中，時間長度 21 之 LSTM 模型之人數證候比例測試集誤差最小。(3)選用神經元個數 32、16、時間長度 21 之 LSTM 模型，與另外兩種天氣因子數量單因子(平均氣壓、平均氣溫)、全因子進行人數證候比例之誤差及證候人數分類之準確度比較，得出單因子之平均氣溫在人數證候比例測試集誤差最小，訓練集則為全因子最小，而在證候人數分類之準確度中，訓練集之準確度以全因子最大，而測試集之準確度以單因子之平均氣溫最大。

關鍵字：中醫證候、氣象因子、深度學習、模型預測

Predictive Analysis of TCM Syndromes and Meteorological Changes Based on Python Deep Learning

Student: Chia-Lun Shih

Advisor: Hung-Pin Chiu, Ph.D.

Department of Information Management

Nanhua University

Master Thesis

ABSTRACT

In recent years, various fields have begun to consider meteorological factors in their implementation strategies, and meteorological economics, meteorological medicine, and other meteorological-related matters have emerged. Traditional Chinese medicine has attached great importance to the relationship between weather and physical changes since ancient times. All causes are the external factors of wind, cold, heat, dampness, dryness and fire corresponding to the body. In TCM, the main diagnostic method is syndrome differentiation and treatment. Each syndrome includes the combination of external causes and internal organs of the body. Therefore, the main objective of this research is to find a suitable deep learning model based on the number of neurons, the length of time, the data type and the number of weather factors.

The results of the study show:

- (1) The length of time is based on the past 14 days, and the weather factor uses the dual factors of average air pressure and average

temperature as sample data. Through the comparison of the number of different neurons in the 4 groups, it is concluded that the number of neurons in the test set of 32 and 16 people's syndrome ratio is the smallest among the other 3 groups of neurons.

- (2) Select the number of neurons 32, 16 and other time lengths of 7, 21 to compare the three models of DNN, RNN, and LSTM. It is concluded that among the number of neurons 32 and 16, the LSTM model with a time length of 21 has the smallest error in the test set of the proportion of people syndromes.
- (3) The LSTM model with 32 neurons, 16 neurons and the length of time 21 was used to compare with the other two kinds of weather factors, single factor (mean air pressure, mean air temperature) and total factor. The results show that the error of single factor average temperature is the smallest in the number of syndrome proportion test set, while the error of all factors is the smallest in the training set. In the accuracy of syndrome number classification, the accuracy of training set is the highest in total factor, while the accuracy of test set is the highest in single factor average temperature.

Keywords: TCM syndromes, Meteorological Factors, Deep Learning, Model Prediction

目錄

論文著作財產權同意書.....	I
誌謝.....	II
摘要.....	III
ABSTRACT.....	V
目錄.....	VII
圖目錄.....	IX
表目錄.....	XII
第一章 緒論.....	1
1.1 研究動機.....	1
1.2 研究目的.....	3
第二章 文獻探討.....	4
2.1 中醫證候與辯證論治.....	4
2.2 氣象因子與氣象醫學.....	5
2.3 Python.....	6
2.4 Keras.....	7
2.5 Spyder.....	7
2.6 深度學習 (Deep Learning).....	7
第三章 研究方法.....	11
3.1 研究架構.....	11
3.2 資料來源.....	13
3.3 資料預處理.....	13
3.4 中醫門診資料篩選.....	13
3.5 中醫辨證系統.....	14
3.6 氣象因子篩選.....	15
3.7 運用工具.....	16
3.8 TimeseriesGenerator().....	16
第四章 資料分析.....	18
4.1 前置動作.....	18
4.2 資料採樣.....	25
4.3 建立與訓練深度學習模型.....	28
4.4 雙因子預測比較.....	30
4.5 單因子預測比較.....	55
4.6 全因子預測比較.....	61
4.7 單因子、雙因子、全因子預測比較.....	63

第五章 結論.....	65
5.1 研究結果.....	65
5.2 研究限制.....	66
5.3 未來發展與建議.....	66
參考文獻.....	67



圖目錄

圖 1	基本深度神經網路.....	8
圖 2	循環神經網路.....	9
圖 3	長短期記憶.....	10
圖 4	研究架構.....	12
圖 5	中醫人數百分比資料讀檔.....	19
圖 6	氣象因子資料讀檔.....	19
圖 7	項目名稱及筆數.....	20
圖 8	輸出平均氣壓及平均溫度.....	20
圖 9	平均氣壓及平均溫度的曲線.....	21
圖 10	平均溫度及平均氣壓標準化.....	22
圖 11	平均溫度及平均氣壓標準化曲線.....	22
圖 12	肝火上炎輸出.....	23
圖 13	肝火上炎曲線.....	23
圖 14	累加後人數百分比資料.....	24
圖 15	累加後人數百分比曲線.....	24
圖 16	時間長度 14 資料採樣.....	26
圖 17	length 採樣範圍.....	26
圖 18	sampling_rate.....	26
圖 19	樣本資料與目標資料.....	27
圖 20	stride 結果.....	27
圖 21	batch_size 結果.....	27
圖 22	取未來第 7 筆資料.....	27
圖 23	訓練集與測試集轉換.....	28
圖 24	DNN 模型.....	29
圖 25	RNN 模型.....	29
圖 26	LSTM 模型.....	29
圖 27	訓練 DNN.....	30
圖 28	訓練 RNN.....	30
圖 29	訓練 LSTM.....	30
圖 30	預測訓練集與測試集.....	31
圖 31	DNN 時間長度 14 神經元個數 24、12 訓練集.....	31
圖 32	DNN 時間長度 14 神經元個數 24、12 測試集.....	32
圖 33	RNN 時間長度 14 神經元個數 24、12 訓練集.....	32
圖 34	RNN 時間長度 14 神經元個數 24、12 測試集.....	33

圖 35	LSTM 時間長度 14 神經元個數 24、12 訓練集.....	33
圖 36	LSTM 時間長度 14 神經元個數 24、12 測試集.....	34
圖 37	神經元個數 24、12 訓練集與測試集損失與誤差.....	34
圖 38	DNN 時間長度 14 神經元個數 32、16 訓練集.....	35
圖 39	DNN 時間長度 14 神經元個數 32、16 測試集.....	35
圖 40	RNN 時間長度 14 神經元個數 32、16 訓練集.....	36
圖 41	RNN 時間長度 14 神經元個數 32、16 測試集.....	36
圖 42	LSTM 時間長度 14 神經元個數 32、16 訓練集.....	37
圖 43	LSTM 時間長度 14 神經元個數 32、16 測試集.....	37
圖 44	神經元個數 32、16 訓練集與測試集損失與誤差.....	38
圖 45	DNN 時間長度 14 神經元個數 48、24 訓練集.....	38
圖 46	DNN 時間長度 14 神經元個數 48、24 測試集.....	39
圖 47	RNN 時間長度 14 神經元個數 48、24 訓練集.....	39
圖 48	RNN 時間長度 14 神經元個數 48、24 測試集.....	40
圖 49	LSTM 時間長度 14 神經元個數 48、24 訓練集.....	40
圖 50	LSTM 時間長度 14 神經元個數 48、24 測試集.....	41
圖 51	神經元個數 48、24 訓練集與測試集損失與誤差.....	41
圖 52	DNN 時間長度 14 神經元個數 64、32 訓練集.....	42
圖 53	DNN 時間長度 14 神經元個數 64、32 測試集.....	42
圖 54	RNN 時間長度 14 神經元個數 64、32 訓練集.....	43
圖 55	RNN 時間長度 14 神經元個數 64、32 測試集.....	43
圖 56	LSTM 時間長度 14 神經元個數 64、32 訓練集.....	44
圖 57	LSTM 時間長度 14 神經元個數 64、32 測試集.....	44
圖 58	神經元個數 64、32 訓練集與測試集損失與誤差.....	45
圖 59	DNN 時間長度 7 神經元個數 32、16 訓練集.....	46
圖 60	DNN 時間長度 7 神經元個數 32、16 測試集.....	46
圖 61	RNN 時間長度 7 神經元個數 32、16 訓練集.....	47
圖 62	RNN 時間長度 7 神經元個數 32、16 測試集.....	47
圖 63	LSTM 時間長度 7 神經元個數 32、16 訓練集.....	48
圖 64	LSTM 時間長度 7 神經元個數 32、16 測試集.....	48
圖 65	DNN 時間長度 21 神經元 32、16 訓練集.....	49
圖 66	DNN 時間長度 21 神經元 32、16 測試集.....	49
圖 67	RNN 時間長度 21 神經元 32、16 訓練集.....	50
圖 68	RNN 時間長度 21 神經元 32、16 測試集.....	50
圖 69	LSTM 時間長度 21 神經元 32、16 訓練集.....	51
圖 70	LSTM 時間長度 21 神經元 32、16 測試集.....	51
圖 71	時間長度 7 神經元 32、16 訓練集與測試集損失與誤差.....	52
圖 72	時間長度 21 神經元 32、16 訓練集與測試集損失與誤差.....	52

圖 73	進行 Kmean 分群.....	54
圖 74	LSTM 準確率訓練及評估.....	54
圖 75	LSTM 雙因子準確率.....	55
圖 76	混淆矩陣.....	55
圖 77	混淆矩陣顯示結果.....	55
圖 78	平均氣壓訓練集預測.....	56
圖 79	平均氣壓測試集預測.....	56
圖 80	平均氣壓損失與誤差.....	57
圖 81	平均氣溫訓練集預測.....	57
圖 82	平均氣溫測試集預測.....	58
圖 83	平均氣溫損失與誤差.....	58
圖 84	平均氣壓準確度.....	59
圖 85	平均氣溫準確度.....	59
圖 86	平均氣壓混淆矩陣.....	60
圖 87	平均氣溫混淆矩陣.....	60
圖 88	全因子訓練集預測.....	61
圖 89	全因子測試集預測.....	62
圖 90	全因子損失與誤差.....	62
圖 91	全因子準確度.....	62
圖 92	全因子混淆矩陣.....	63

表目錄

表 1	中醫門診資料來源篩選表.....	14
表 2	氣象因子篩選表.....	15
表 3	TimeseriesGenerator()參數.....	16
表 4	4 組神經元測試集比較.....	45
表 5	三種模型不同時間長度的比較.....	53
表 6	平均氣壓與平均氣溫誤差比較表.....	59
表 7	單因子類別比較.....	60
表 8	單、雙、全因子比較表.....	64



第一章 緒論

1.1 研究動機

在這資訊爆炸成長的時代，各種資料不斷被創造及累積，透過整合這些不同領域的大數據並萃取出資料的核心，運用資料分析獲取可被利用的價值，轉化成知識後，擬定行銷策略，可協助企業做出好的決策。以氣象資訊而言，日本的 7-ELEVEN 運用 POS (Point of Sales) 銷售時點情報系統分析氣象因素找出的「七五三感冒指數」是最著名的「氣象經濟」，他們發現如果一天當中溫度相差七度、今天和昨天的溫度差五度，濕度大於百分之三十，那麼感冒的人數就會增加。可以考慮把感冒藥、體溫計和口罩放在貨架上。在這個氣象經濟時代，把握溫度變化就等於把握商機(楊蕙菁，2004)。

除了氣象經濟外，氣象醫學這幾年也逐漸受到重視，是研究氣候變化對人類健康和疾病影響的科學。《氣候變遷與人類健康》(賴昂廷、林益卿、楊鈺雯、吳美鳳，2012)中提到全球氣候變化與人類健康的關係是多樣、複雜和重要的，建立預警系統及時掌握當地最新天氣動態，可以減少天氣對人體健康的威脅。

從數千年前開始，中醫藥即非常重視氣候變化對人體所導致的影響。黃帝內經就從基本養生開始說起，如《內經·四氣調神大論》說:逆春氣，則少陽不生，肝氣內變。逆夏氣，則太陽不長，心氣內洞。逆秋氣，則太陰不收，肺氣焦滿。逆冬氣，則少陰不藏，腎氣獨沈。此外，以六十年為一周期的運氣醫學中討論天氣與疾病的關連性，如《內經·歲露論》就特別提到:人與天地相參也，與日月相應也(張志聰，1670)。以上皆說明人要

懂得順應四季的時節，同時也要合於每時期的氣象變化，如此明白氣象變化和人體相對應的關係，乃是養生、保健與治療的關鍵重點。

中醫最重要的就是辨證論治。有了「證型」或「證候」則可推演出相關的中醫處方及針灸選穴，推拿按摩的處置方法。而中醫證型是依疾病症狀的關聯所推演出來的結果，其中包括中醫的病位、病因與病機而組合出來中醫的「證」。例如，社區型獲得性肺炎就可分成十三種證型，例如風熱犯肺，痰熱壅肺，痰濕阻肺，肺脾氣虛，肺陰虛等等證型(張洪春、李得民、齊蕊，2019)。而準確的辨證是分析中醫證型與氣象因素的關鍵。大林慈濟醫院中醫部及中正大學資工所經過多年在中醫症狀標準化及詞庫的研究，建立一套中醫辨證系統(陳信賢，2015)，系統辨證結果顯示，依照書本上案例分析，大約 86%-97%是與《中醫診斷學》等教科書一致，最高準確度為 97%。

癌症已經蟬聯多年國人十大死因的榜首，因此，本研究想要利用大林慈濟醫院中醫門診資料中醫證型的變化與地區氣象變異之間的關係，探討在特定氣象因素之下，是否有出現癌症病患中醫證型的變化特徵，以提供雲嘉地區醫療人員更進一步瞭解本地病患的罹病的特質。本研究的資料來源是大林慈濟醫院 2010 年至 2013 年的中醫門診資料庫，本研究將對其病歷記錄進行前處理，並且以中正大學資工所所開發的中醫辨證系統，進行自動化辨證，以建立中醫證型的時序資料庫。同時也以網路爬蟲抓取中央氣象局 2010 年至 2013 年的雲嘉南地區氣象資料，擷取六氣所牽涉到的 16 個氣象因子，以建立氣象因素時序變異資料庫。藉由此二種時序資料庫的整合分析，探討中醫證型與氣象變動資料的關聯性。

1.2 研究目的

本研究將以深度學習技術，來探討中醫證型與地區氣候變異的預測模型。本研究的目的如下：

1. 建立中醫診斷時序資料庫與氣象資訊時序資料庫。
2. 使用深度學習技術建立氣象因素與證候人數百分比變化的預測模型。
3. 將數值型資料轉成類別型資料進行類別準確度的評估。
4. 對數值預測與類別準確度進行比較。



第二章 文獻探討

本章將彙整中醫證候與辨證、氣象醫學、深度學習等文獻進行探討與研究，藉由探討這些文獻驗證本研究之基礎架構，達到有憑有據之合理探討，以建構後續系統。

2.1 中醫證候與辨證論治

中醫治病講究「辨證求因，審因論治」。「證」就是「證型」或「證候」，代表當時身體的巨觀狀態。一個證型背後有套病因病機、與該病因病機所引發的一組症狀。證候是病機的外在表現，病機是證候的內在本質。由於病機的內涵中包括了病變的部位、原因、性質和邪正盛衰變化，故證候能夠用來說明人體的不平衡以及對抗致病因素的能力，揭示了病變的機理和發展趨勢。例如，風寒束表、肝陽上亢、心血虧虛、心脈痹阻等，都是證候的概念。

中醫治證候，少直接針對症狀。一個證候只反映整個疾病發展過程中的一個縱切面，亦即人體這個巨系統一時的態(state)、模型(model)。而一個疾病的發展過程，可視為許多證候的串接。根據症狀，辨別證候，推斷疾病發生的原因，從而採取相應的治療方法，便顯示出中醫治療疾病的特色：「辨證論治」。

辨證，即是四診（望、聞、問、切）所收集的有關疾病的所有資料，包括症狀和體徵，運用中醫學理論進行分析、綜合，辨清疾病的原因、部位、性質及發展趨向，然後概括、判斷為某種性質之證候的過程。由於證候是疾病過程中某一階段或某一類型的病理概括，只能反映疾病某一階段和某一類型的病變本質，故中醫學在辨識證候時，要求同時辨明疾病的病

因、病位、病性及病勢，即辨明疾病從發生到轉歸的總體病機(張慈文, 2017)，由此可見，「辨證論治」為中醫在診斷與治療疾病時，考量所有面向以取得平衡對應的重要觀念與過程。

2.2 氣象因子與氣象醫學

氣象因素是指導致天氣發生變化的各種因素，常見的有氣溫、雨量、風速、氣壓，以及四季相對應的氣候。氣象因子常用的氣溫數值為日溫差、月均溫、年均溫、年溫差，雨量的常用數值為年雨量、年平均雨量、降雨強度、雨量變率，風速的常用數值為季風、颱風、焚風、落山風，氣壓的常用數值為低氣壓或高氣壓。氣象因素對於我們的食、衣、住、行、育、樂及生、老、病、死都存在著相關的影響，也是間接導致結果發生的原因，若能配合氣象因子調整生活作息，反而是一種預防疾病的方法。有學者透過資料採礦找出氣候及環境因子對心血管疾病之影響因子，並且建立預測模型，其研究結果可做為潛在心血管病患預防醫療上的參考與建議(萬擎等人, 2016)。

中醫認為人體疾病的發生發展與自然界周期性的節微變化有密切關係。每當節氣轉變時，氣鬱多而久病之人，交節前三日多死。而腠理乾塞，交節不能過，交節後死為主(史桂榮, 2011)。「六氣學說」為黃帝內經整理先人所累積的豐富醫療經驗，以此為基礎形成系統化之理論，對於氣象與人體之間擁有一套推算規則，其中六氣為風、寒、暑、溼、燥、火(胡建勳、劉堯欽, 2008)，中醫的「六氣學說」實為氣象醫學的先驅。中醫研究也提到類風溼關節炎患者在冬季發病機率較大，表示寒冷氣候會加劇類風溼關節炎患者人數(楊孝兵等人, 2014)。

氣象因素，包括溫度、濕度、相對濕度、大氣壓力、風速，也會對不同中醫證型的骨痺患者有影響，其中秋末和冬季發病人多，證型上以腎虛髓虧、陽虛寒凝和瘀血阻滯患者發病人次均與溫度、濕度及氣壓有顯著負相關(黃彥等人，2013；周熙等人，2013)。這兩篇也是少數有提到中醫證型與氣象因素有關的論文。而在呂小琴等人的研究中，以 2166 例流感病例的中醫證型統計發現 2016-2017 年冬季流感發生受平均溫度影響，但不同病原的中醫證型間無明顯區別，都以風熱證為主(劉清泉等人，2018)，但若是統計時間拉長至十年以上的話，可能會有不同的結果。

氣象因素也會影響成人社區獲得性肺炎中醫證型的發生，特別有相關的是風熱犯肺與痰熱壅肺與平均日照時數、平均風速、平均氣溫及相對濕度及平均降雨量有關(黃晶一、朱威、顏仕星、周紅，2018)。在出血性中風或是缺血性中風的發病，均存在明顯的季節分布特徵，高氣壓，低氣壓和低氣溫的的條件都是好發時間(武月萍，2007)。

利用 Pearson 關聯係數與單因子變異數分析，研究某月特定證候的癌症病患人數與該月前一段時間的氣候變化之相關性。其研究結果顯示心脈痺阻(氣滯)、肝火上炎與肝陽上亢等證候和均溫有正相關，肝火上炎證候和氣壓有負相關，而心脈痺阻(寒凝)證候和平均風速有負相關者(陳宗仁，2019)。

2.3 Python

Python 由 Guido van Rossum 創造，是一種功能強大且被廣泛使用的直譯式程式語言。Python 強調的是簡潔的語法與程式碼的可讀性，相比於 Java 或 C++，Python 讓開發者能夠用更少的程式碼來表達想法。無論是小型或是大型程式開發，Python 都試圖讓程式的結構更為清晰明瞭。

2.4 Keras

Keras 是由 François Chollet 所開發，是一個用 Python 語言所編寫的函式庫，可以快速地實現深度學習，將模型的輸入層、隱藏層、輸出層，做好架構，只需在神經元、激活函數等參數加入到架構中。由於 Keras 非常好用，TensorFlow 將 Keras 納入自己的函式庫中，命名為 tf.keras。

2.5 Spyder

Spyder 是一個使用 python 語言的整合開發環境(Integrated Development Environment, IDE)，擁有的資源有 Matplotlib、NumPy、IPython 等，提供了代碼編輯、交互測試等特性，支持的系統有 Linux、Windows 等，除了一般 IDE 有的調適器、編輯器，還擁有歷史命令視窗、變數查看器、互動式命令視窗等組件。

2.6 深度學習(Deep Learning)

深度學習為近年從機器學習所衍伸出的領域，透過機器模擬人腦的運作方式，促使其機器就像人類一樣具備學習的能力。機器學習需要人為做特徵擷取分類，而深度學習則自己到資料庫內比對特徵後，將特徵放置到多層神經元的類神經網路 (Neural Networks) 裡，將隱藏層與輸出層使用網路架構連接，選擇適合的網路架構做深度學習訓練，從中得出模型 (陳奕廷，2016；羅賢君等人，2018)。如今，有各種各樣的深度學習框架，如深度神經網路、卷積神經網路、循環神經網路等。這些框架已應用於語

音辨識、音訊識別、電腦視覺、自然語言處理等，與傳統方法相比，有著更好的效果。

2.6.1 深度神經網路(Deep Neural Networks, DNN)

DNN 根據不同層的位置進行劃分。神經網路層可以分為三種類型，輸入層、隱藏層和輸出層，如(圖 1)。輸入數據從輸入層加權後，傳遞到隱藏層，再傳遞到輸出層，計算出最終輸出與訓練數據中目標值的誤差。也可以使用反向傳遞，以梯度下降法來訓練、調整神經網路參數，使損失函數最小化。

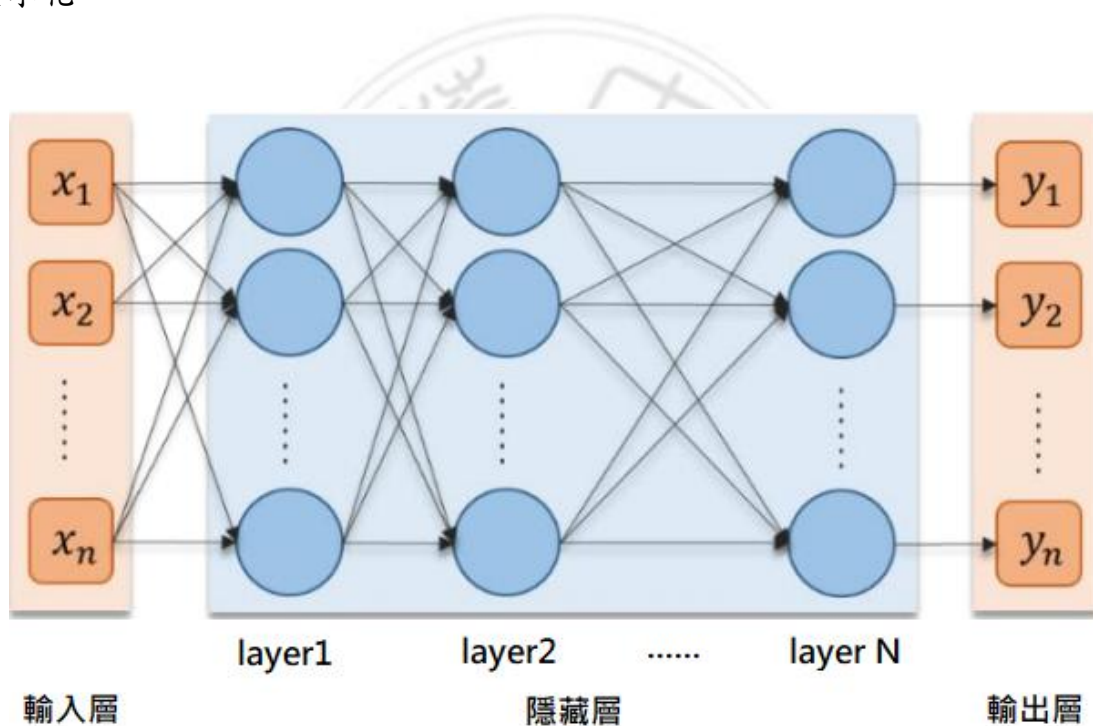


圖 1 基本深度神經網路

2.6.2 循環神經網路(Recurrent Neural Network, RNN)

RNN 多了時間的概念，可以按照時間序列將這些資訊建立關聯性，透過循環特性，當資料按照時間順序輸入時，如果資料在循環中還沒輸入完，

當前的輸出會負責記憶，這種輸出會稱為狀態，接下來的狀態會按照過往的記憶決定結果，循環結束後，最終輸出會負責預測，如(圖 2)。

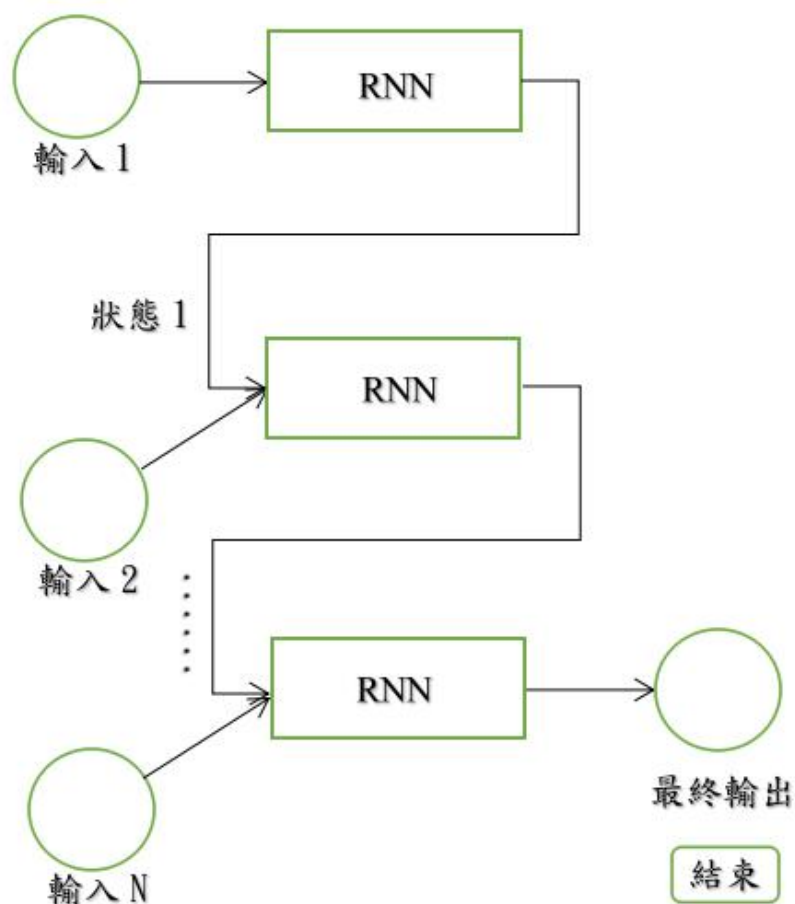


圖 2 循環神經網路

2.6.3 長短期記憶(Long Short-Term Memory, LSTM)

LSTM 是一種 RNN。RNN 的狀態就像短期記憶，隨著時間的傳遞，短期記憶的資訊會被遺忘，為了解決此問題，Hochreiter 和 Schmidhuber 提出了 LSTM 的概念，LSTM 多了一個乘載記憶的空間，可以讓產生出的狀態暫存，讓 RNN 能存放更多的記憶，如(圖 3)。

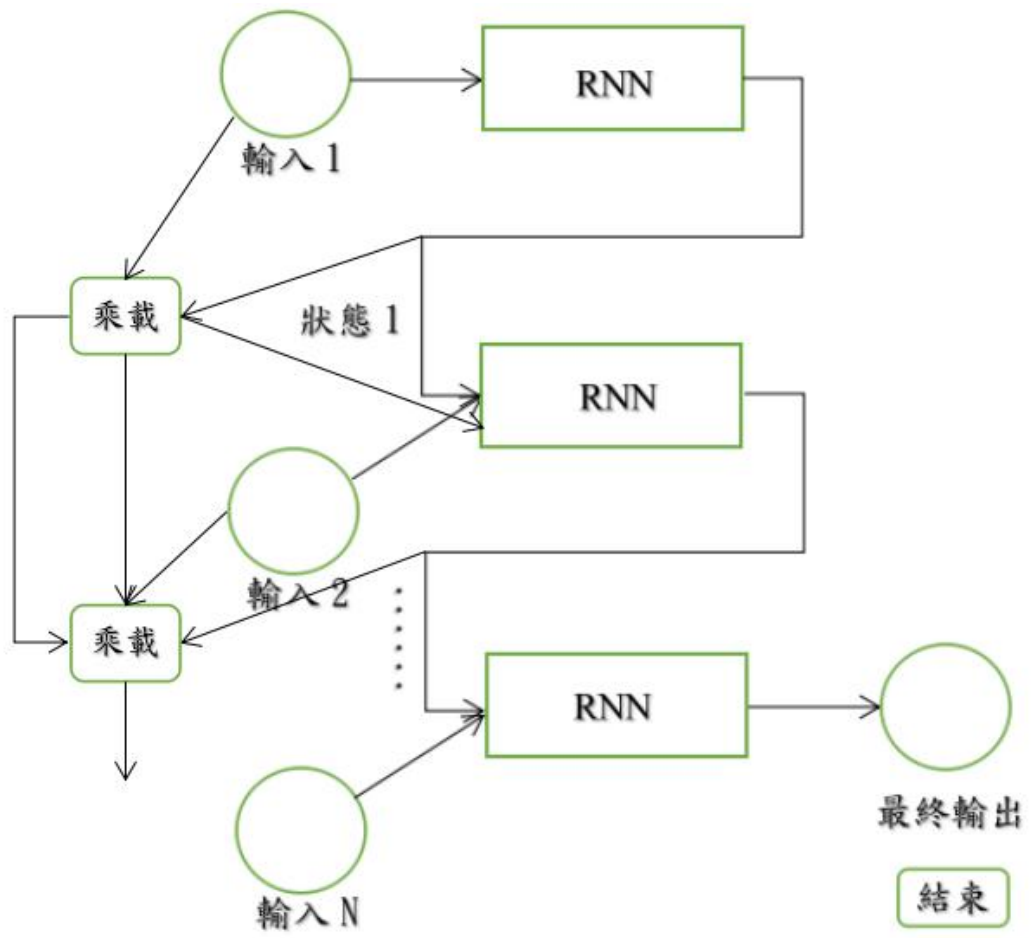


圖 3 長短期記憶

第三章 研究方法

本章根據前兩章研究背景、文獻探討，總結歸納出研究架構。本章將分為八個部分，3.1 為研究架構、3.2 為資料來源，3.3 為資料預處理，3.4 為中醫門診資料篩選，3.5 為中醫辨證系統，3.6 為六氣之氣象因子篩選，3.7 為運用工具，3.8 為 TimeseriesGenerator()。

3.1 研究架構

本研究將氣象因子資料和中醫門診資料進行篩選與預處理，再對氣象因子進行每個時間序列資料之標準化，以使用相似之基準取得相對小的數值。建立 TimeseriesGenerator()，將天氣資料做為樣本資料，中醫資料作為目標資料，產生訓練集與測試集。資料採樣後，建立 DNN、RNN、LSTM 三種深度學習模型，進行訓練、編譯、預測後，比較三種模型之效果，最後將中醫資料從數值型轉換成類別型，對類別型模型進行評估。本研究之研究架構如(圖 4)所示。

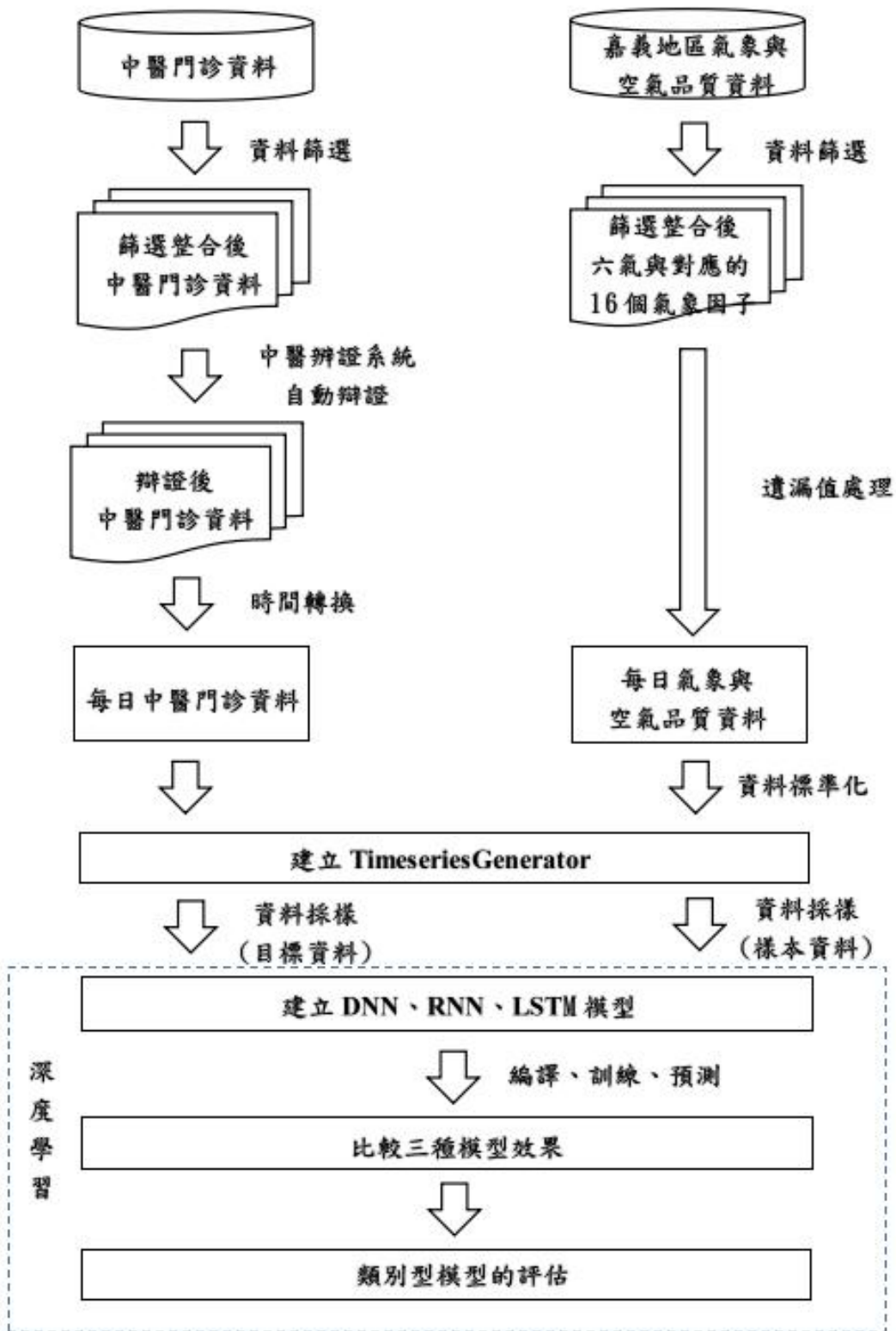


圖 4 研究架構

3.2 資料來源

本研究中醫門診資料是從大林慈濟醫院中醫部取得 2010 年到 2013 年的病歷資料，氣象資料與空氣品質資料則是利用 Python 網路爬蟲將中央氣象局嘉義地區氣象站氣象公開資料及行政院環境保護署之環境資源資料擷取下來，並將所取得的資料轉換整合成適合分析的資料檔。

3.3 資料預處理

基於上述操作，氣象因子資料和中醫門診資料已有原始資料，分別針對兩份資料進行篩選與預處理，由於空氣品質和氣象資料資料是以感測器來偵測，感測器可能會出現失靈等狀況，導致特定某些時段並無資料紀錄，在進行深度學習前，發現最大降水量、累積降水量、累積降水時數等氣象因子資料有遺漏值，為避免之後的分析出現誤差，以 python 計算出每項氣象因子資料的平均值進行填補。為本研究所需，需將資料的型態進行轉換，以方便後續之分析。

3.4 中醫門診資料篩選

中醫門診數據來自嘉義大林慈濟醫院中醫部。原始門診數據從 2000 年 1 月至 2013 年 12 月開始累積。本研究篩選出 2010 年 1 月至 2013 年 12 月居住在雲嘉南地區的地區患者的資料，再將欄位進行整合，如表 1 所示，總共有 38,608 筆資料。

表 1 中醫門診資料來源篩選表

欄位	欄位目的
性別	病人基本資訊
年齡	
就醫日期	用於累計一段時間內的病歷數量
主述	病患所描述之症狀
客觀	經由醫師判斷之症狀
ICD-9 代碼	病患明確症狀代碼
住址	用來對應氣候資料的地理位置

3.5 中醫辨證系統

上述數據過濾後的數據未經轉換，不適合數據分析，且原始中醫門診數據沒有證候欄位，故採用中正大學研究團隊開發的中醫辨證系統，將數據轉化為合適之中醫證候，目前共整理出虛證 20 種和實證 47 種。

中醫辨證系統將證候對應的原始症狀標準化，轉化為標準症狀，然後進行自動化辨證，並根據模糊理論計算出各證候的歸屬度分數。由於系統只接受中文，如果出現英文，會將英文替換為空字串，標點符號則會作為分詞依據，替換為中醫辨證系統能夠識別的分詞符號。最後，通過中醫辨證系統進行成功辨證的可用數據總計 19065 筆病歷。

中醫資料為每日就診之資料，每日就診資料會存在單筆或多筆，為配合每日氣象因子的格式，將就診資料以日為單位進行整合，共有 1461 筆。

就診人數逐年上升，如果直接以每日人數進行分析，會有誤差的產生，以百分比對就診人數進行轉換，將每日證候人數除上該年證候總人數，得出該日患有此證候之人數百分比，以利後續深度學習使用。

$$\text{就診人數百分比} = \frac{\text{證候人數(天)}}{\text{證候總人數(年)}} \times 100\%$$

3.6 氣象因子篩選

氣象因子資料庫內之資料分別為從中央氣象局與行政院環境保護署中所取得之原始資料，以嘉義地區 2010 年 1 月至 2013 年 12 月之氣候與空氣品質資料為主。本研究將依據中醫學說之外感六淫(六氣)為基礎，針對 48 個氣象因子欄位進行篩選，篩選出 15 個適合的氣象因子欄位，最後，透過計算氣象區間之最高氣溫減去最低氣溫得出溫差，總共 16 個欄位，如表 2 所示。

表 2 氣象因子篩選表

六氣	判斷依據	篩選欄位
風	皮膚為空氣中致病因子進入身體之部位，故選擇會與皮膚接觸之欄位。	平均風速(m/s)
		測站平均氣壓(hPa)
		測站最高氣壓(hPa)
		測站最低氣壓(hPa)
		PM10(μg/m ³)
		PM2.5(μg/m ³)
寒	選擇與寒冷或炎熱特性有關之欄位。	平均氣溫(°C)
暑		最高氣溫(°C)
		最低氣溫(°C)
火		累積日照時數(hr)

六氣	判斷依據	篩選欄位
溼	選擇與潮濕或乾燥特性有關之欄位。	平均相對溼度(%)
		累積降水量(mm)
累積降水時數(hr)		
燥		累積降水日數(天)
		最大日降水量(mm)

3.7 運用工具

本研究使用工具為 Spyder，利用 TensorFlow 內以 Python 做為開發語言的 Keras 建立深度學習模型。

3.8 TimeseriesGenerator()

TimeseriesGenerator() 是 Keras 用來處理時序資料時的工具，可以對連續的時序資料進行採樣，並批次處理、生成標籤資料與樣本資料。當使用 TimeseriesGenerator() 時，會建立一個 Sequence 物件，需傳入處理方式和資料，而批次資料不會在建立 Sequence 物件時就產生，需要在下一批次資料產生後或是走訪 for 迴圈時，才從原資料產生。TimeseriesGenerator() 的參數，如(表 3)所示。

表 3 TimeseriesGenerator() 參數

參數	參數說明
data	採樣的時序資料。
targets	目標資料，時序長度必須與 data 一致。

參數	參數說明
length	採樣窗口長度，每次要取多長的資料進行採樣。
sampling_rate	每次採樣要隔幾個資料點，預設為 1。
stride	採樣窗口間要隔幾個資料點，預設為 1。
start_index	資料採樣的起始點。
end_index	資料採樣的結束點。
shuffle	打亂資料的順序，預設為 False，為不打亂。
reverse	是否要反向採樣，預設為 False。
batch_size	批次量大小。



第四章 資料分析

本研究以 DNN 為基礎，建立深度學習模型。中醫證候與氣象因子之資料具有時序性，故第二個模型採用多了時間概念之 RNN。本研究會透過不同時間長度進行比較，當時間長度很長時，RNN 因輸出張量有限，難以保存較多的資訊，故第三個模型採用比 RNN 更能存放更多記憶之 LSTM。

證候以肝火上炎為主要目標，依據陳宗仁(2019)的研究結果顯示，平均氣溫與平均氣壓對肝火上炎有相關，故本研究先將平均氣溫與平均氣壓雙因子作為天氣因子，透過不同神經元個數與不同時間長度，建立 DNN、RNN 及 LSTM 深度學習預測模型，並將所得出之研究結果進行比較、結果討論與分析。

4.1 前置動作

對每日中醫就診人數百分比資料及每日氣象因子資料進行讀檔，如(圖 5)與(圖 6)所示。因深度學習不需項目名稱進行分析及讀檔到最後一列時會多出一筆空白資料，故將這兩列刪除。最後輸出資料有兩種資料的項目名稱及兩種資料的筆數各 1461 筆，如(圖 7)所示。


```

# %% 讀取證候人數百分比資料

pname = 'the_day_people.csv'
p = open(pname, encoding = 'cp950') # ←使用"cp950"編碼來讀檔
data2 = p.read()
p.close()

lines2 = data2.split('\n')
header2 = lines2[0].split(',') # ←第一列資料為項目名稱
lines2 = lines2[1:] # ←取項目名稱之外的資料
del lines2[len(lines2)-1] # ←最後一筆會空白，所以刪除

#print(lines2)
print(len(lines2))
print(header2)

```

圖 5 中醫人數百分比資料讀檔

```

# %% 讀取天氣資料

wname = 'the_day_weather.csv'
w = open(wname, encoding = 'cp950') # ←使用"cp950"編碼
data = w.read()
w.close()

lines = data.split('\n')
header = lines[0].split(',') # ←第一列資料為項目名稱
lines = lines[1:] # ←取項目名稱之外的資料
del lines[len(lines)-1] # ←最後一筆會空白，所以刪除

print(len(lines))
print(header)

```

圖 6 氣象因子資料讀檔

```

1461
['觀測時間(day)', '測站氣壓(hPa)', '測站最高氣壓(hPa)', '測站最低氣壓(hPa)', '氣溫(°C)',
'最高氣溫(°C)', '最低氣溫(°C)', '溫差', '風速(m/s)', '降水量(mm)', '降水時數(hour)', '相
對溼度(%)', '日照時數(hour)', 'PM10(懸浮微粒)', 'PM2.5(細懸浮微粒)']
1461
['日期', '心氣虛', '心血虛', '心陰虛', '心陽虛', '肝血虛', '肝陰虛', '脾氣虛', '脾陽虛',
'脾氣陷', '脾不統血', '胃陰虛', '肺氣虛', '肺陰虛', '腎精不足', '腎不納氣', '腎氣不固',
'腎陰虛', '腎陽虛', '心陽暴脫', '大腸陽虛', '風襲衛表', '寒襲衛表', '暑襲衛表', '濕襲衛
表', '燥襲衛表', '熱襲衛表', '風襲肌膚', '濕襲肌膚', '風襲經絡', '寒襲經絡', '風襲關節',
'濕襲關節', '心脈痺阻(寒凝)', '心火亢盛', '心脈痺阻(痰聚)', '痰火擾心(鬱)', '痰火擾心
(躁)', '痰火擾心(外感熱)', '痰迷心竅', '心脈痺阻(血瘀)', '心脈痺阻(氣滯)', '小腸實熱', '寒
邪客肺', '熱邪壅肺', '痰濕阻肺', '痰熱壅肺', '大腸濕熱', '大腸液虧', '大腸實熱', '寒邪客
脾', '濕邪客脾', '熱邪客脾', '寒邪客胃', '熱邪客胃', '胃氣上逆', '食滯胃脘', '肝陽化風',
'陰虛動風', '血虛生風', '熱極生風', '寒滯肝脈', '肝膽濕熱', '肝火上炎', '肝陽上亢', '肝氣
鬱結', '膽郁痰擾', '膀胱濕熱']

```

圖 7 項目名稱及筆數

取出除了日期之外之全部氣象因子，將 1461 筆資料轉換成 Numpy 陣列後，並利用 matplotlib 將資料的曲線繪製出來，也可分別取出因子，如(圖 8)所示，取出雙因子分別為平均氣壓及平均氣溫。(圖 9)為繪製出平均氣壓及平均溫度的曲線，其中藍色曲線為平均氣壓，橘色曲線為平均氣溫。

```

In [149]: temp_data = raw_data[:, 0:4:3]#取天氣資料(平均氣壓、均溫)
...: print(temp_data)
[[1014.7  16.3]
 [1012.9  17.6]
 [1012.6  17.7]
 ...
 [1017.1  13.3]
 [1016.9  15.3]
 [1015.6  16.3]]

```

圖 8 輸出平均氣壓及平均溫度

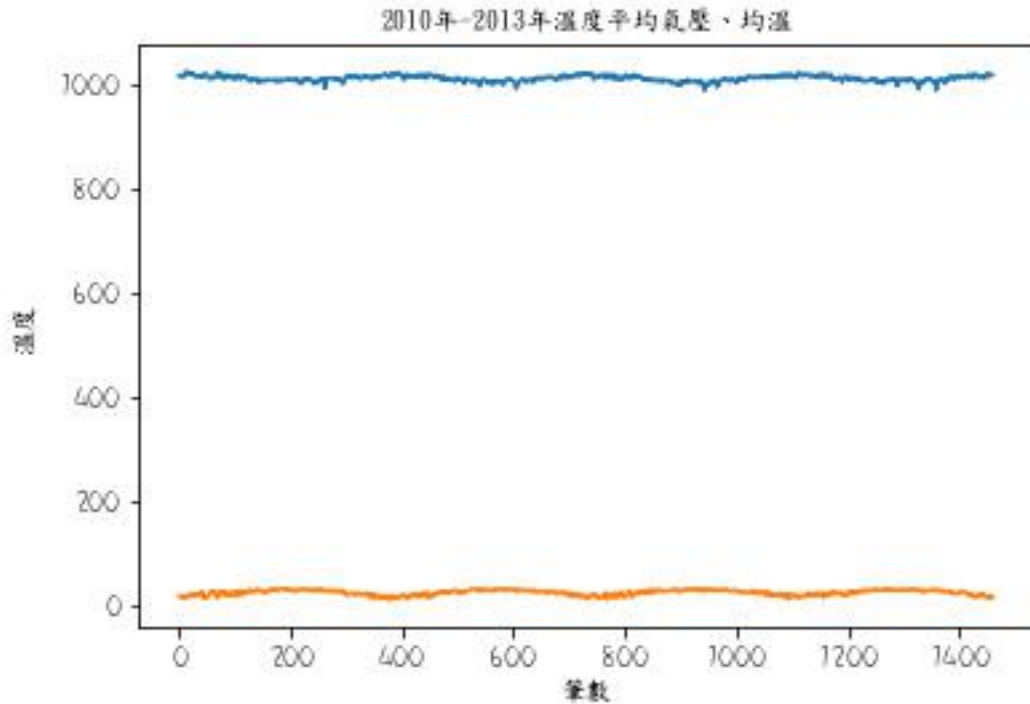


圖 9 平均氣壓及平均溫度的曲線

將資料預先處理成神經網路可以讀取的格式，但天氣資料中的平均氣溫及平均氣壓都有不同的基準，需獨立標準化這兩個時間序列資料，以便可以用相似的基準取得相對小的數值，每筆資料減去平均值並除以標準差來預先處理資料。因本研究只使用 2010 年 1 月至 2012 年 12 月共 1096 個時間點作為訓練資料，故只計算此部分資料之平均值與標準差，避免資訊洩漏給模型，發生成效較佳的假象，如(圖 10)所示。取得數值後，繪製出標準化後的平均溫度及平均氣壓的曲線圖，藍色曲線為標準化後之平均氣壓，橘色曲線為標準化後之平均氣溫，如(圖 11)所示，可看出平均氣壓及平均氣溫在一年之中之變化及週期性。

```
## 資料標準化
mean = temp_data[:1096].mean(axis=0)
temp_data -= mean
std = temp_data[:1096].std(axis=0)
temp_data /= std
plt.plot(temp_data)
plt.show()
```

圖 10 平均溫度及平均氣壓標準化

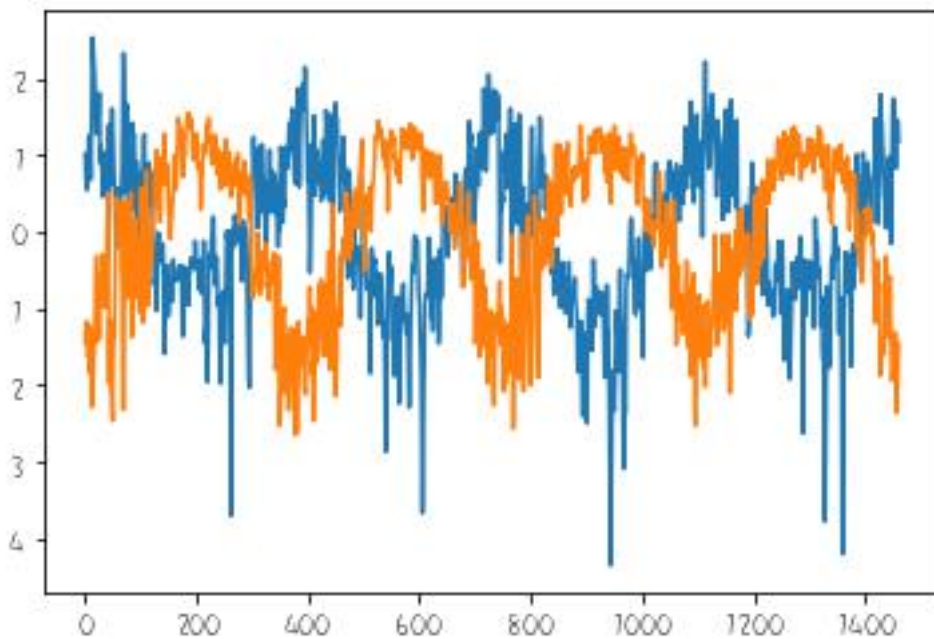


圖 11 平均溫度及平均氣壓標準化曲線

(圖 12)取出肝火上炎資料，將 1461 筆資料轉換成 Numpy 陣列後輸出，並使用 matplotlib 將資料的曲線繪製出來，如(圖 13)所示。因肝火上炎每日就診資料並不是每日皆有就診人數，為考量此情況，取第一筆資料時，將此筆資料與未來 7 天資料進行累加(圖 14)，並繪製成曲線圖，如(圖 15)所示。

```
In [52]: people_data = np.zeros((len(lines2), len(header2) - 1))
...: for i, line in enumerate(lines2):
...:     values = [float(x) for x in line.split(',')[1:]]#不放日期，只有天氣資料
...:     #print(values)
...:     people_data[i, :] = values
...:     #print(raw_data)
...:
...:
...: people_data = people_data[:, 62]#取人數資料(肝火上炎)
...: print(people_data)
...: #全部筆數
...: plt.plot(range(len(people_data)), people_data)
...: plt.show()
[0.  0.  0.  ... 0.  0.4199 0.5399]
```

圖 12 肝火上炎輸出

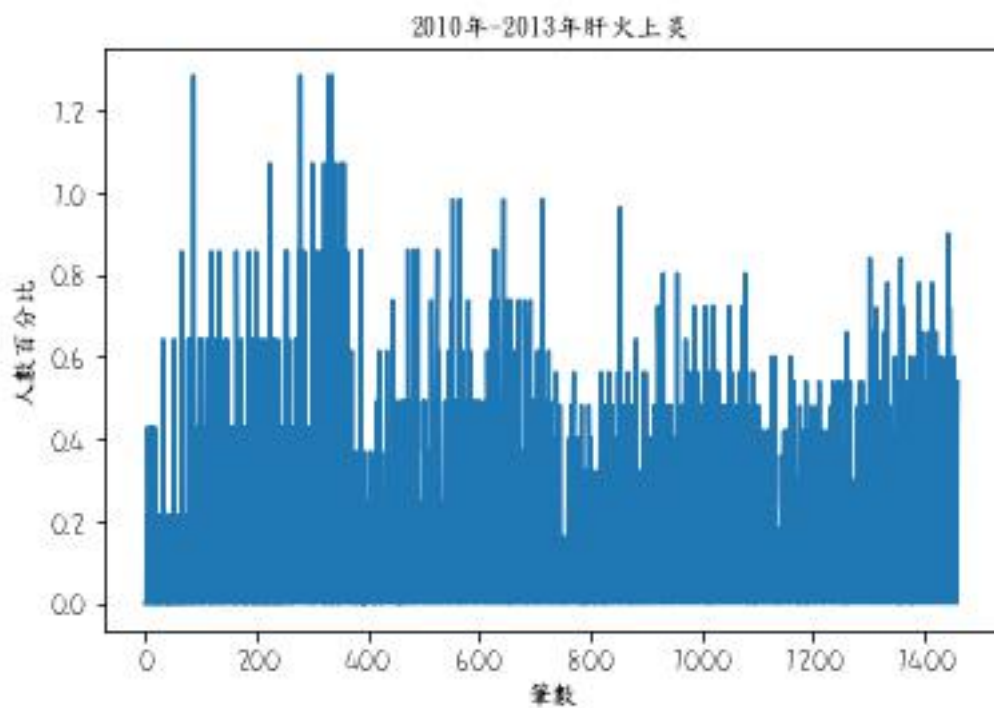


圖 13 肝火上炎曲線

```

#將未來7天平均
temp = people_data.copy()

for i in range(0, len(people_data)):
    sum = 0
    for j in range(i, min(i+7, len(people_data))):
        sum += people_data[j]
    temp[i] = sum
#print(people_data[:10])
#print(temp[:10])
people_data = temp
print(len(people_data))

```

圖 14 累加後人數百分比資料

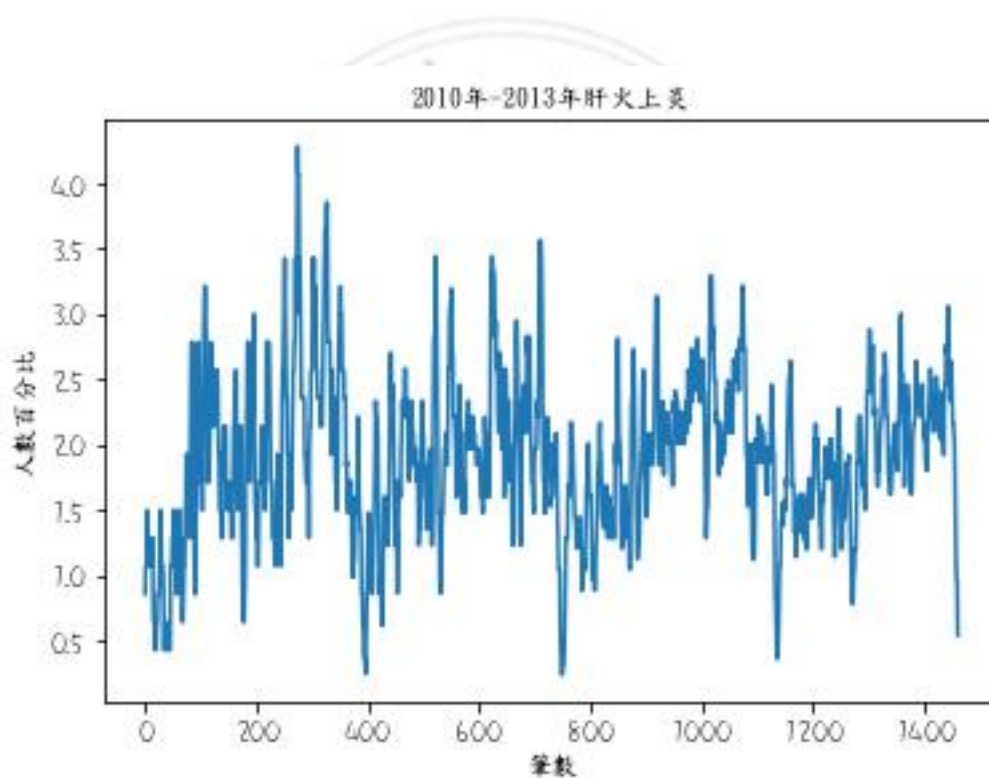


圖 15 累加後人數百分比曲線

4.2 資料採樣

本研究將以過去 7 天、過去 14 天及過去 21 天三種不同時間長度，去預測未來 7 天後的人數百分比。(圖 16)以過去 14 天為範例，樣本資料(data)為每日之平均氣溫及平均氣壓，目標資料(target)為累加 7 天之肝火上炎每日人數百分比。(圖 17)以時間長度 length 選出要採樣的資料範圍，由於 reverse 預設為 False，故以最左邊的第一個資料為第一個採樣點。(圖 18)採樣到的資料會加入到樣本資料的陣列中，下一次會隔 1 個資料點 (sampling_rate)。(圖 19)當採集框內無資料可採集時，則此筆樣本資料採集完畢，而採集框外的第一筆為目標資料。(圖 20)第一筆樣本資料與目標資料出來後，會將採樣框向右移 1 個單位(stride)。(圖 21)重複以上動作直到完成 6 筆資料(batch_size)，就完成第一個批次的採樣。

若想取未來第 7 筆資料，而非取下一筆資料時，需先將樣本資料截掉後 6 筆、目標資料截掉前 6 筆，取目標時就會取到下一筆加截掉筆數之資料，如(圖 22)所示。最後產生訓練資料與測試資料。

```

from tensorflow.keras.preprocessing.sequence import TimeseriesGenerator

length = 14
delay = 7 #要取未來的第7筆資料
sampling_rate = 1
stride = 1
batch_size = 6

train_test = 1096-length-(delay-1)

data = temp_data[:-(delay-1)] #使用容器切片的技巧截掉後6筆資料，作為訓練資料
target = people_data[(delay-1):] #截掉前6筆資料，作為目標資料

#產生訓練資料
train_gen = TimeseriesGenerator(data, target,
                                length=length,
                                sampling_rate=sampling_rate,
                                stride=stride,
                                start_index=0,
                                end_index=train_test-1,
                                batch_size=batch_size)

test_gen = TimeseriesGenerator(data, target,
                                length=length,
                                sampling_rate=sampling_rate,
                                stride=stride,
                                start_index=train_test,
                                end_index=None,
                                batch_size=batch_size)

print(train_gen[0][0].shape) #顯示樣本資料的 shape

```

圖 16 時間長度 14 資料採樣

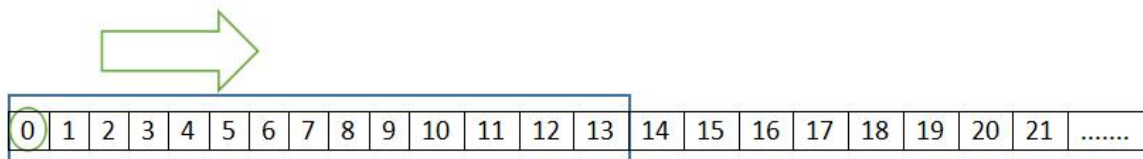


圖 17 length 採樣範圍

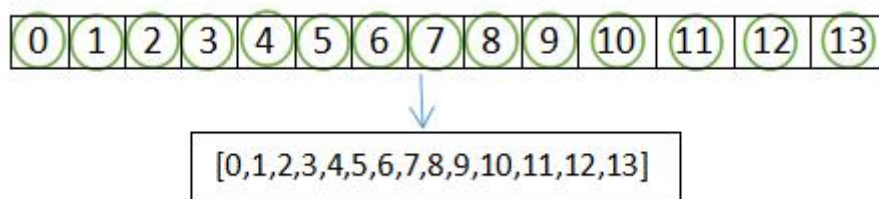


圖 18 sampling_rate

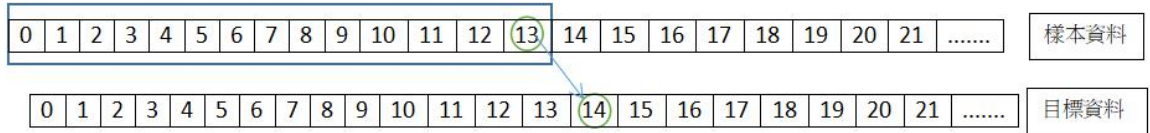


圖 19 樣本資料與目標資料

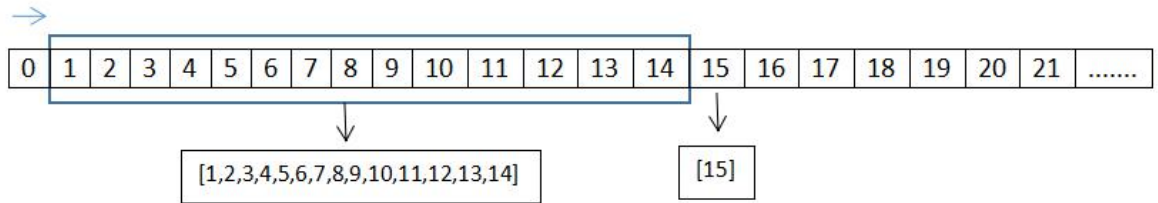


圖 20 stride 結果

樣本資料:
 [0,1,2,3,4,5,6,7,8,9,10,11,12,13],
 [1,2,3,4,5,6,7,8,9,10,11,12,13,14],
 [2,3,4,5,6,7,8,9,10,11,12,13,14,15],
 [3,4,5,6,7,8,9,10,11,12,13,14,15,16],
 [4,5,6,7,8,9,10,11,12,13,14,15,16,17],
 [5,6,7,8,9,10,11,12,13,14,15,16,17,18]

目標資料:
 [14,15,16,17,18,19]

圖 21 batch_size 結果

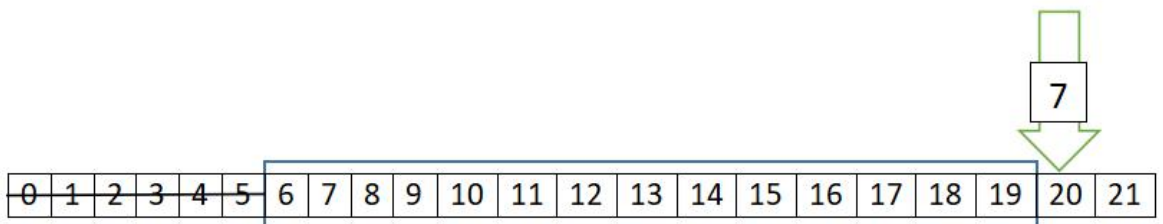


圖 22 取未來第 7 筆資料

將訓練集與測試集的資料依序取出，分別加入一維的串列中，之後轉換成 Numpy 陣列，以便繪製後續模型訓練集與測試集結果曲線圖，如(圖 23)所示。

```
test_temp=[]

for datas in test_gen:
    for temp in datas[1]:
        test_temp.append(temp)  #將肝火上炎訓練集依序取出，並加進 1D 的串列中

test_temp = np.array(test_temp)

train_temp=[]

for datas in train_gen:
    for temp in datas[1]:
        train_temp.append(temp)  #將肝火上炎測試集依序取出，並加進 1D 的串列中
```

圖 23 訓練集與測試集轉換

4.3 建立與訓練深度學習模型

4.3.1 建立深度學習模型

建立 DNN 的第一層需用 Flatten 展平層，將 3 維資料展平，展平層的輸入值 length 為時間長度、feature 為特徵數，再經由三個密集層處理。建立 RNN 的第一層使用 SimpleRNN，透過 input_shape 參數來指定時間長度與特徵數，再經由兩個密集層處理。建立 LSTM 模型與 RNN 類似，第一層不是使用 SimpleRNN，而是使用 LSTM 層。

為了防止過度學習的狀況發生，添加了 Dropout 層，neurons1 與 neurons2 為神經元個數，而神經元個數後面之 relu 啟動函數能有效地克服梯度消失

的問題，亦可緩解過度學習的問題。由於是回歸問題，故最後一個密集層沒有使用的啟動函數，如(圖 24-26)所示。

```
### 建立密集神經網路並訓練

dense_model = Sequential()
dense_model.add(layers.Flatten(input_shape=(length, feature)))

dense_model.add(layers.Dense(neurons1, activation='relu'))
dense_model.add(layers.Dropout(0.2))
dense_model.add(layers.Dense(neurons2, activation='relu'))
dense_model.add(layers.Dense(1))
dense_model.summary()
```

圖 24 DNN 模型

```
### 建立RNN模型、訓練並輸出預測結果

rnn_model = Sequential()
rnn_model.add(layers.SimpleRNN(neurons1, input_shape=(length, feature)))
rnn_model.add(layers.Dense(neurons2, activation='relu'))
rnn_model.add(layers.Dropout(0.2))
rnn_model.add(layers.Dense(1))
rnn_model.summary()
```

圖 25 RNN 模型

```
### 建立LSTM模型、訓練並輸出預測結果

lstm_model = Sequential()
lstm_model.add(layers.LSTM(neurons1, input_shape=(length, feature)))
lstm_model.add(layers.Dense(neurons2, activation='relu'))
lstm_model.add(layers.Dropout(0.2))
lstm_model.add(layers.Dense(1))
lstm_model.summary()
```

圖 26 LSTM 模型

4.3.2 訓練深度學習模型

(圖 27-29)分別對 DNN、RNN、LSTM 三種模型進行訓練，此問題為回歸問題，故 loss 使用均方誤差作為損失函數，metrics 使用平均絕對誤差作為評估函數。損失函數與評估函數通常為反指標，評估成效越好，損失值

就越小。將訓練週期設定為 100 次，每次使用損失函數傳回損失值時，優化器 optimizer 會對其進行優化，而評估函數只會用來評量成效，不會用來訓練。

```
dense_model.compile(optimizer='adam',loss='mse',metrics=['mae'])  
  
dense_history = dense_model.fit(train_gen,  
                                epochs=100, #←設定訓練週期為100次  
                                )
```

圖 27 訓練 DNN

```
rnn_model.compile(optimizer='adam',loss='mse',metrics=['mae'])  
  
rnn_history = rnn_model.fit(train_gen,  
                             epochs=100  
                             )
```

圖 28 訓練 RNN

```
lstm_model.compile(optimizer='adam',loss='mse',metrics=['mae'])  
  
lstm_history = lstm_model.fit(train_gen,  
                               epochs=100  
                               )
```

圖 29 訓練 LSTM

4.4 雙因子預測比較

本研究先以過去 14 天之平均氣壓與平均氣溫雙因子為樣本資料，對肝火上炎人數百分比進行預測，因樣本資料為雙因子，故將特徵數 feature 設為 2。利用神經元個數進行變換，對 DNN、RNN、LSTM 模型進行比較，神經元個數分為 4 組，第一組個數為 24、12，第二組個數為 32、16，第三組個數為 48、24，第四組個數為 64、32。

將訓練集與測試集進行預測後，如(圖 30)，並繪製訓練集與測試集的預測結果圖。

```
prediction = dense_model.predict(train_gen)
prediction = dense_model.predict(test_gen)
```

圖 30 預測訓練集與測試集

4.4.1 神經元個數 24、12

神經元個數 neurons1 與 neurons2 分別為 24、12，並繪製出 DNN、RNN、LSTM 三種模型的訓練集與測試集的預測結果，如(圖 31-36)所示。

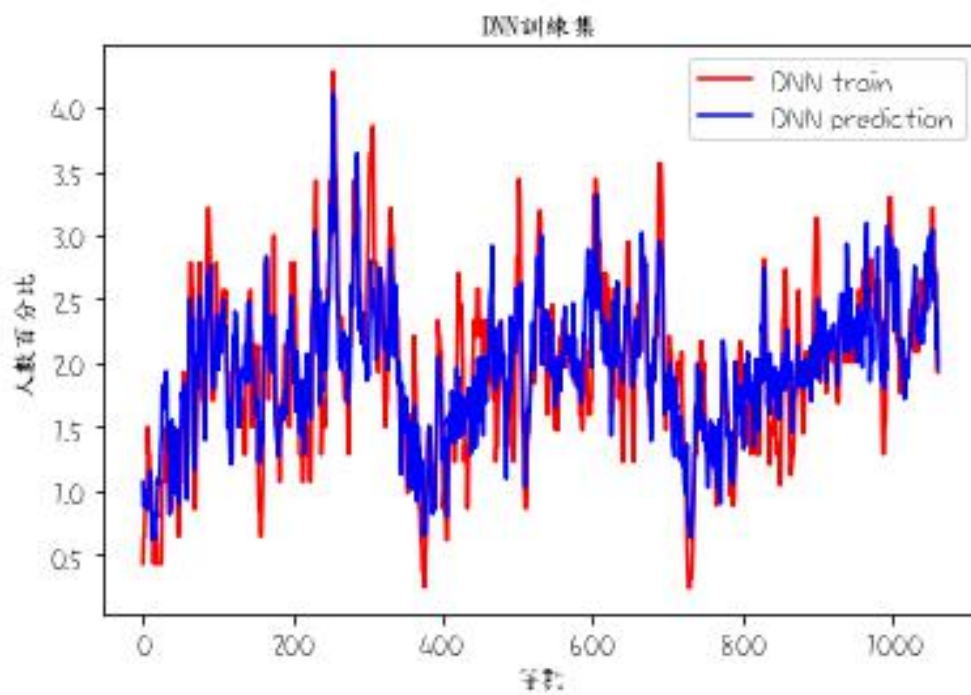


圖 31 DNN 時間長度 14 神經元個數 24、12 訓練集

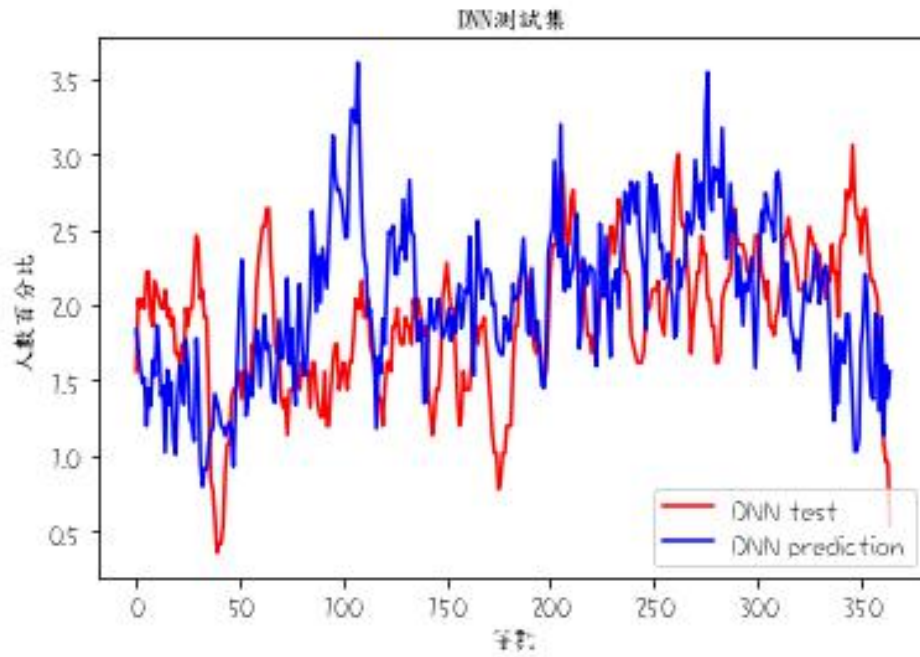


圖 32 DNN 時間長度 14 神經元個數 24、12 測試集

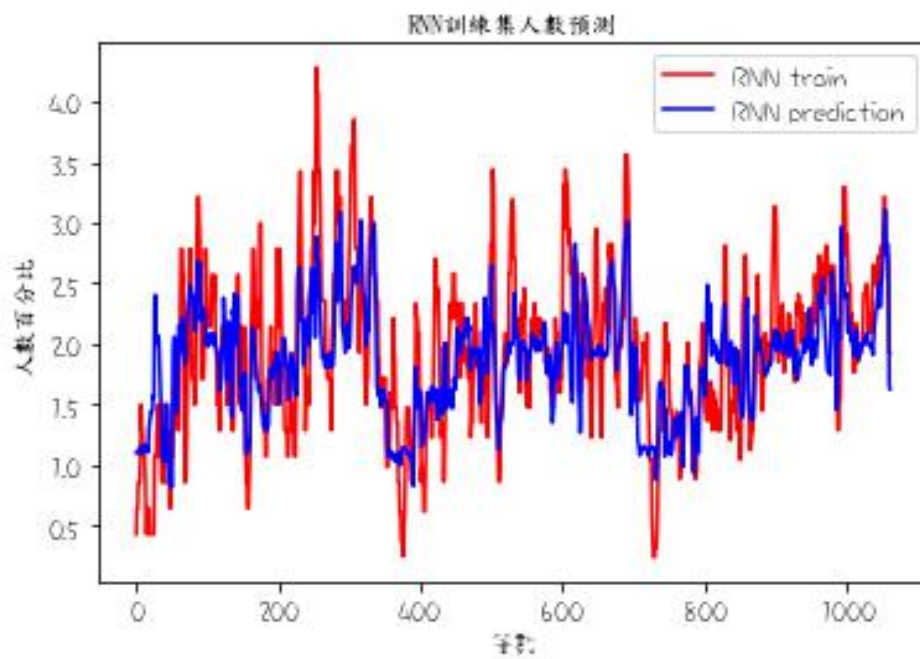


圖 33 RNN 時間長度 14 神經元個數 24、12 訓練集

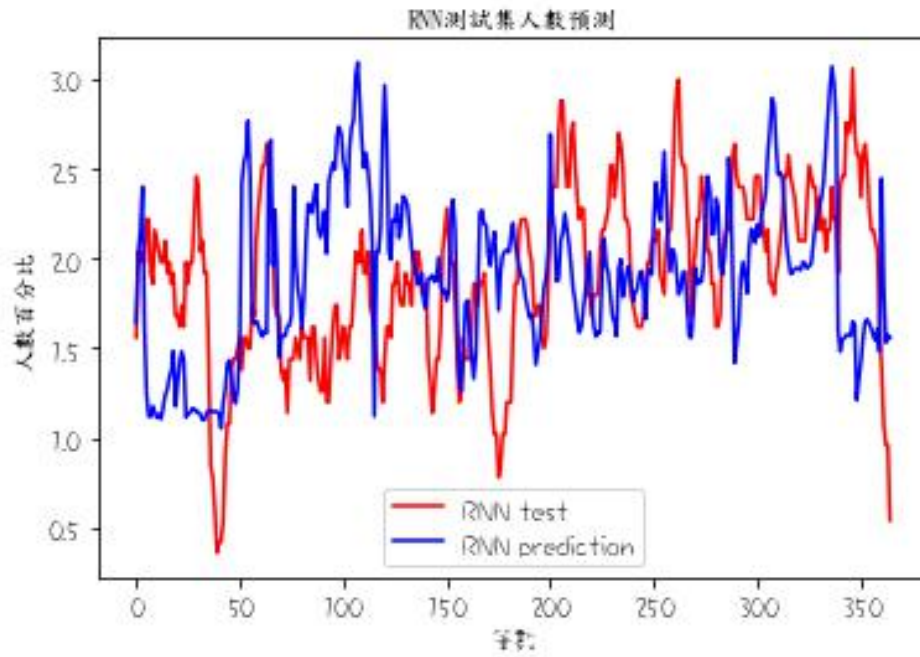


圖 34 RNN 時間長度 14 神經元個數 24、12 測試集

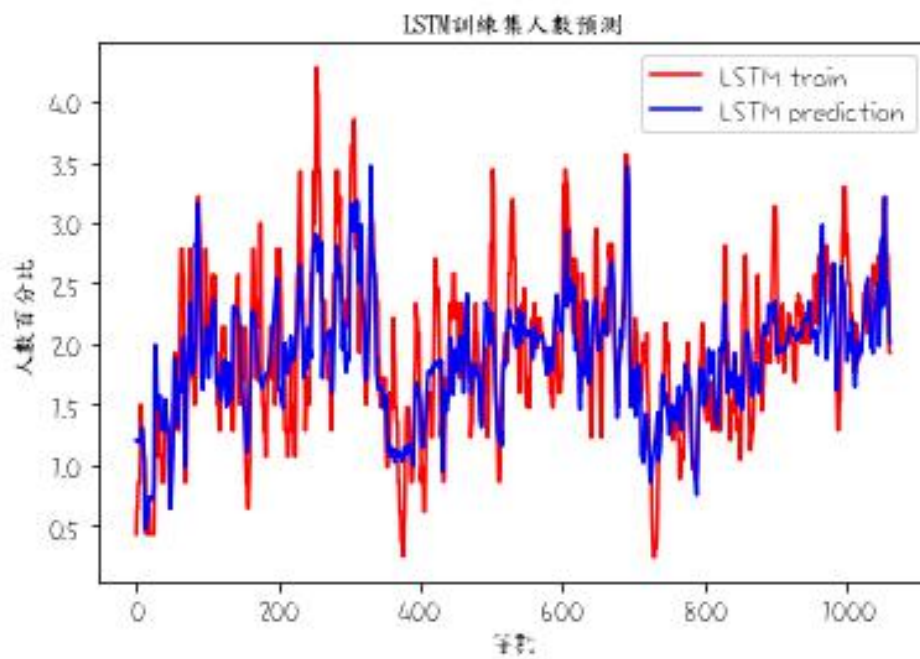


圖 35 LSTM 時間長度 14 神經元個數 24、12 訓練集

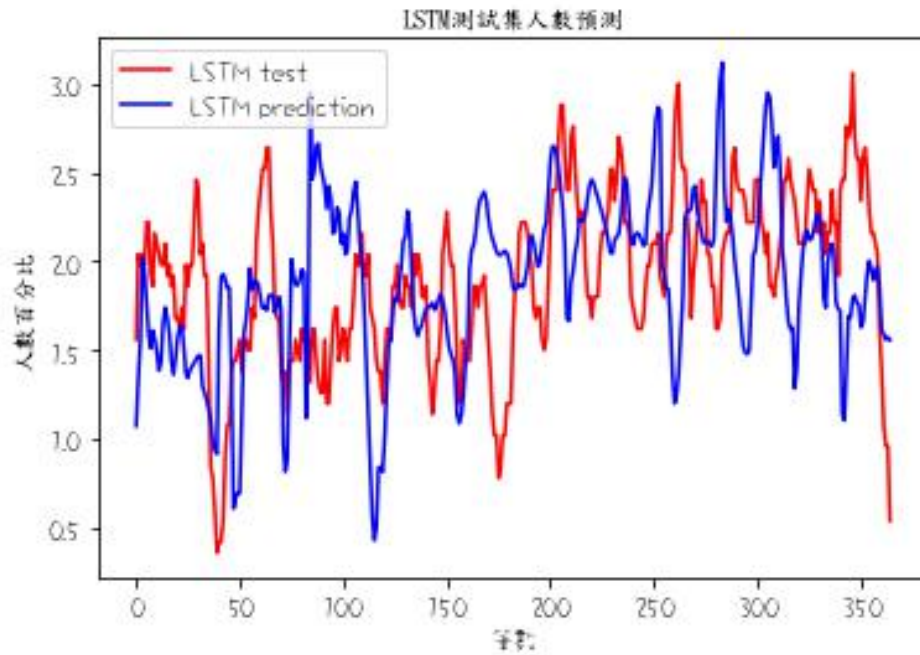


圖 36 LSTM 時間長度 14 神經元個數 24、12 測試集

(圖 37)為三種模型對訓練集與測試集的評估，從測試集來看，LSTM 的誤差為最小，代表神經元個數 24、12 較適合使用 LSTM 的模型來實作。

```
Dense NN訓練集: [0.14145734906196594, 0.29520878195762634]
RNN訓練集: [0.22515173256397247, 0.3720088601112366]
LSTM訓練集: [0.17508620023727417, 0.3293195962905884]
Dense NN測試集: [0.40830662846565247, 0.5264898538589478]
RNN測試集: [0.39095965027809143, 0.5245664715766907]
LSTM測試集: [0.36820879578590393, 0.49794408679008484]
```

圖 37 神經元個數 24、12 訓練集與測試集損失與誤差

4.4.2 神經元個數 32、16

神經元個數 neurons1 與 neurons2 分別為 32、16，並繪製出 DNN、RNN、LSTM 三種模型的訓練集與測試集的預測結果，如(圖 38-43)所示。

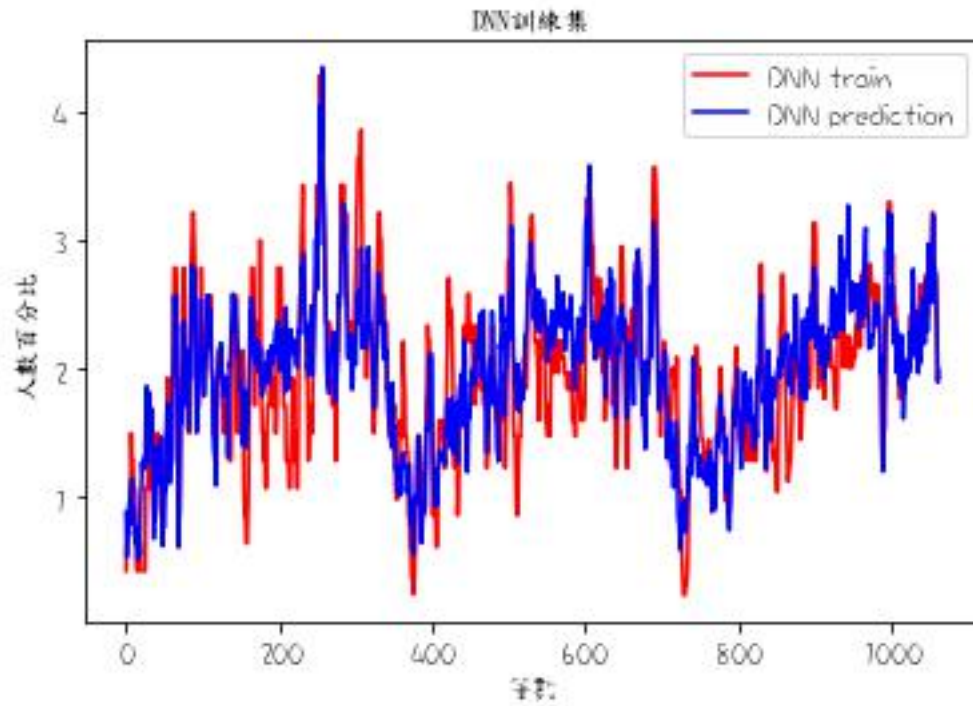


圖 38 DNN 時間長度 14 神經元個數 32、16 訓練集

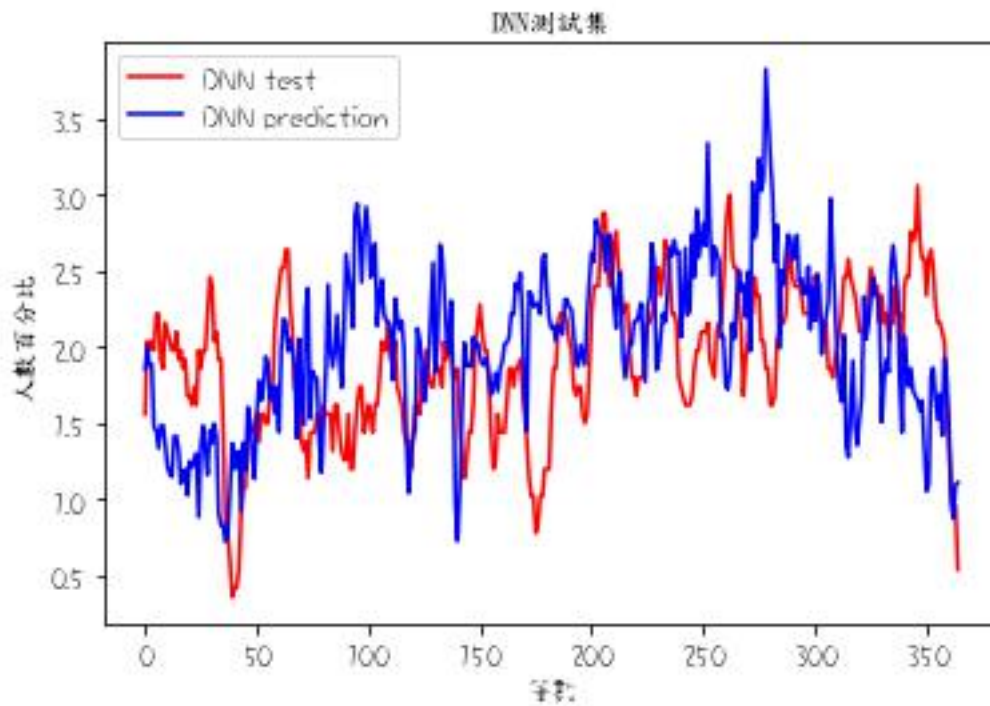


圖 39 DNN 時間長度 14 神經元個數 32、16 測試集

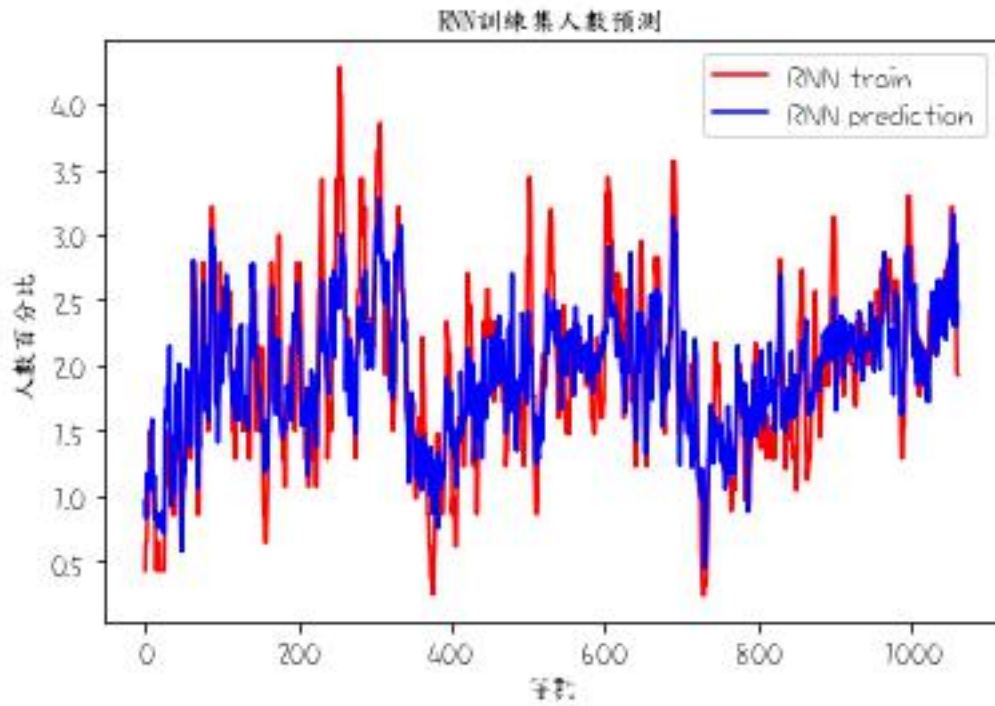


圖 40 RNN 時間長度 14 神經元個數 32、16 訓練集

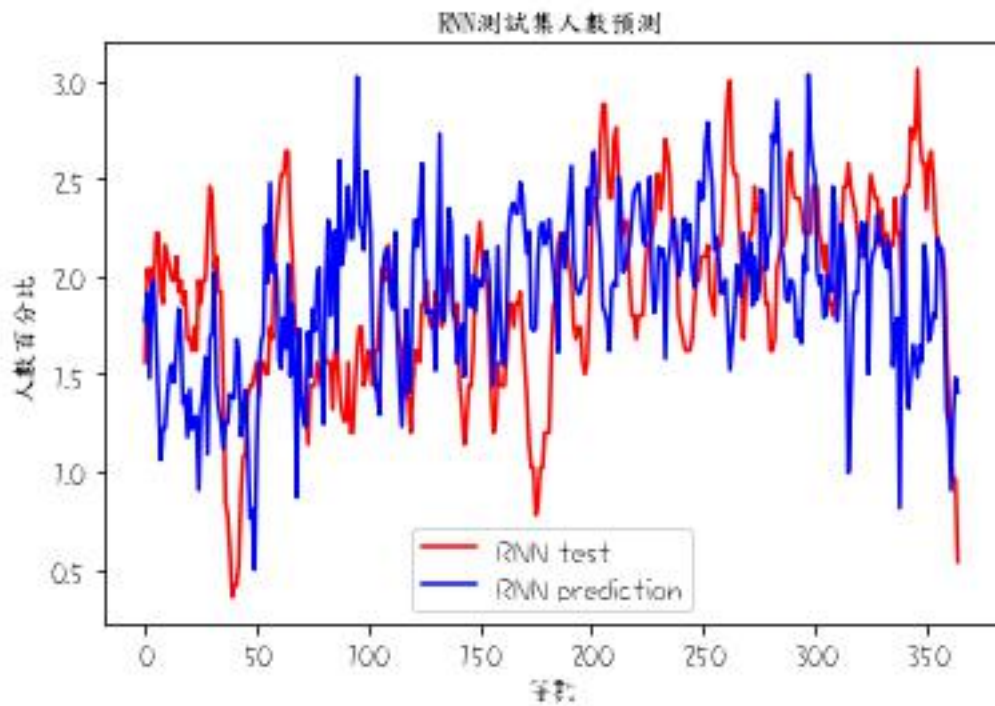


圖 41 RNN 時間長度 14 神經元個數 32、16 測試集

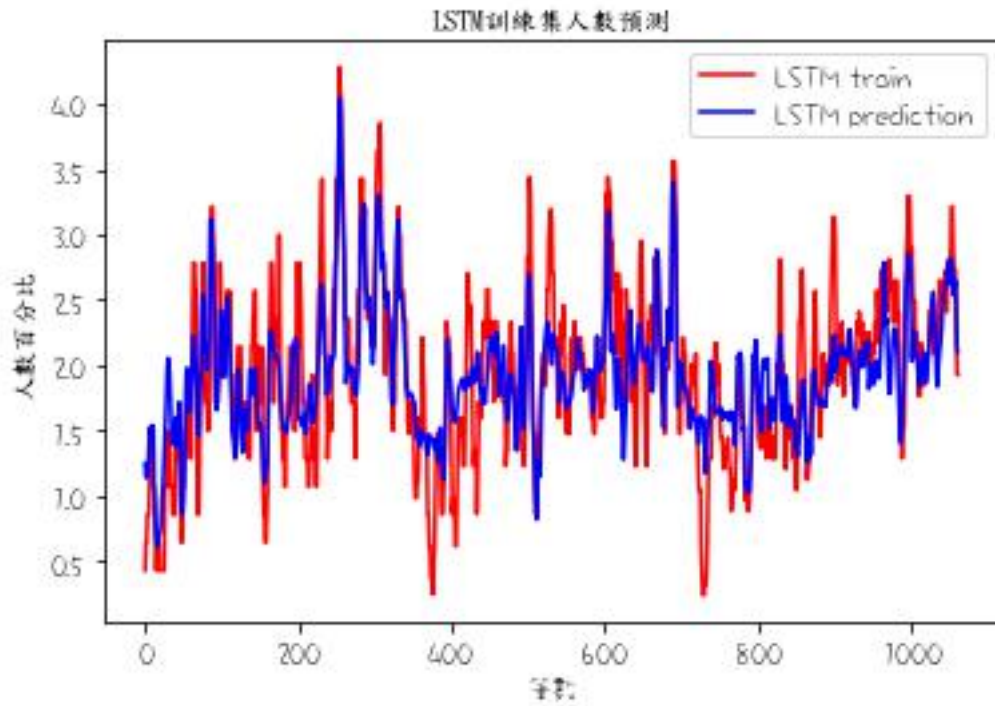


圖 42 LSTM 時間長度 14 神經元個數 32、16 訓練集

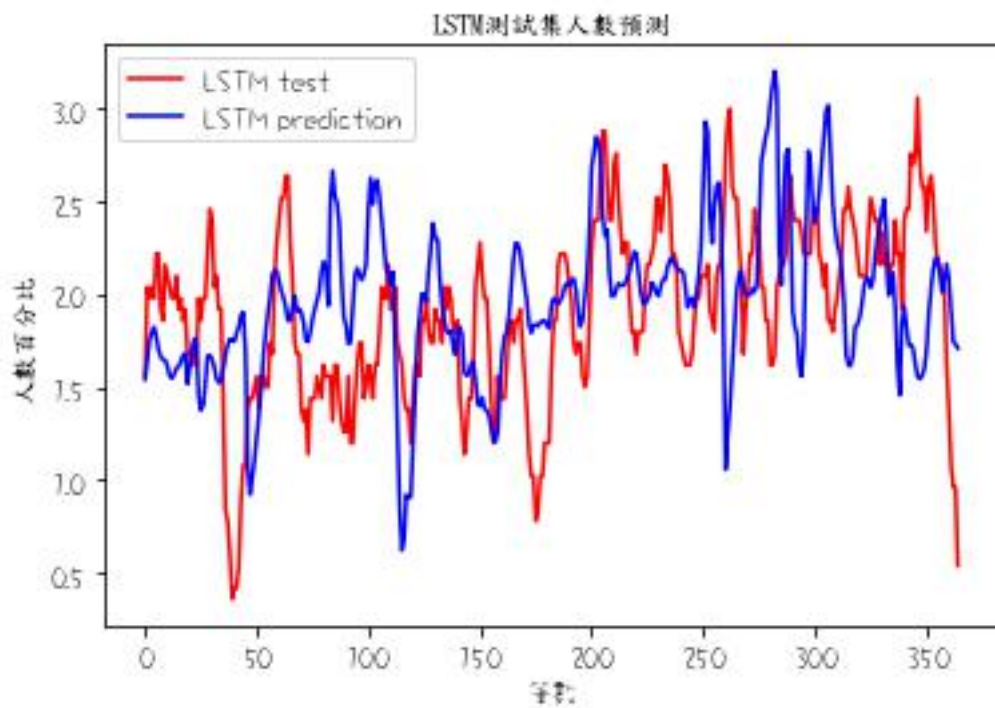


圖 43 LSTM 時間長度 14 神經元個數 32、16 測試集

(圖 44)為三種模型對訓練集與測試集的評估，從測試集來看，LSTM 的誤差為最小，代表神經元個數 32、16 較適合使用 LSTM 的模型來實作。

```
Dense NN訓練集: [0.15703485906124115, 0.3079240918159485]
RNN訓練集: [0.14587372541427612, 0.3040865957736969]
LSTM訓練集: [0.1560356169939041, 0.3048568069934845]
Dense NN測試集: [0.3999325633049011, 0.5037596225738525]
RNN測試集: [0.3327919542789459, 0.46534913778305054]
LSTM測試集: [0.3223455846309662, 0.4581795036792755]
```

圖 44 神經元個數 32、16 訓練集與測試集損失與誤差

4.4.3 神經元 48、24

神經元個數 neurons1 與 neurons2 分別為 48、24，並繪製出 DNN、RNN、LSTM 三種模型的訓練集與測試集的預測結果，如(圖 45-50)所示。

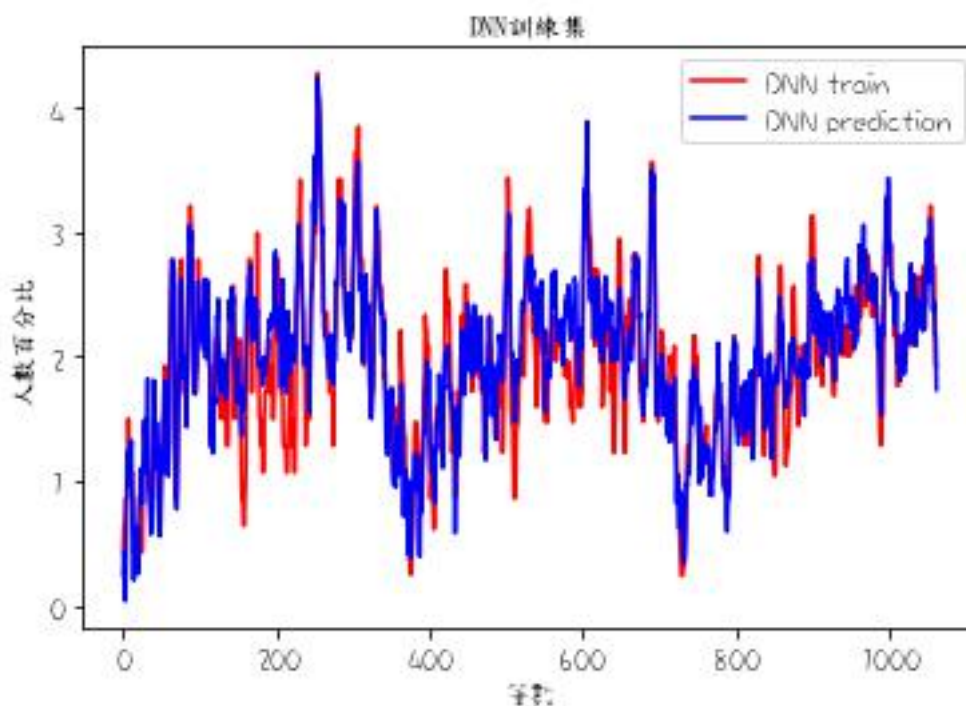


圖 45 DNN 時間長度 14 神經元個數 48、24 訓練集

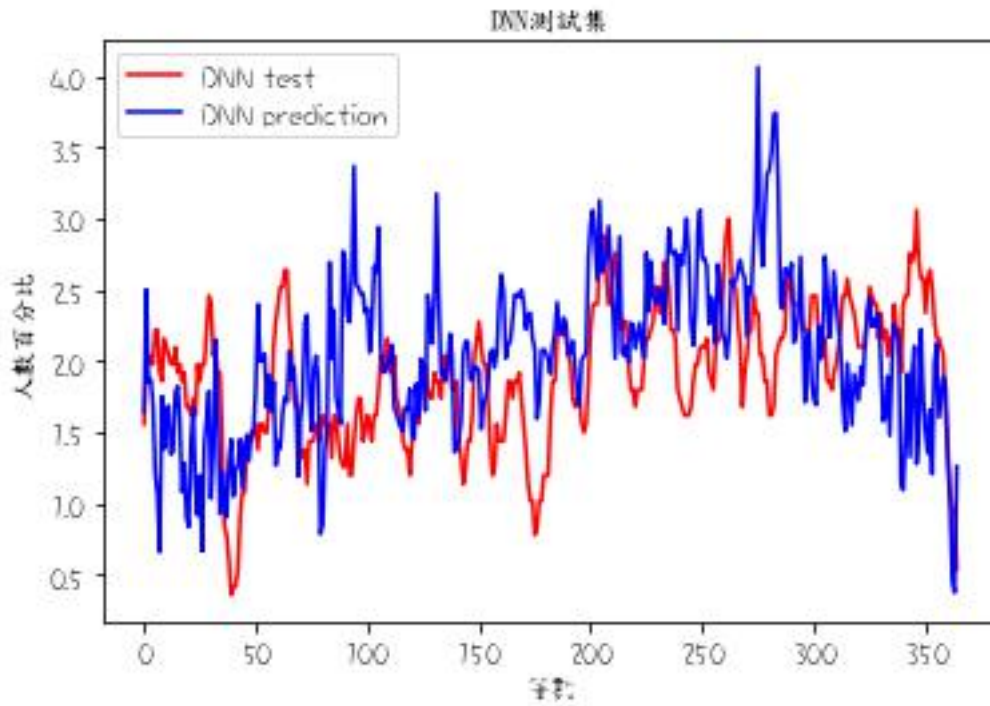


圖 46 DNN 時間長度 14 神經元個數 48、24 測試集

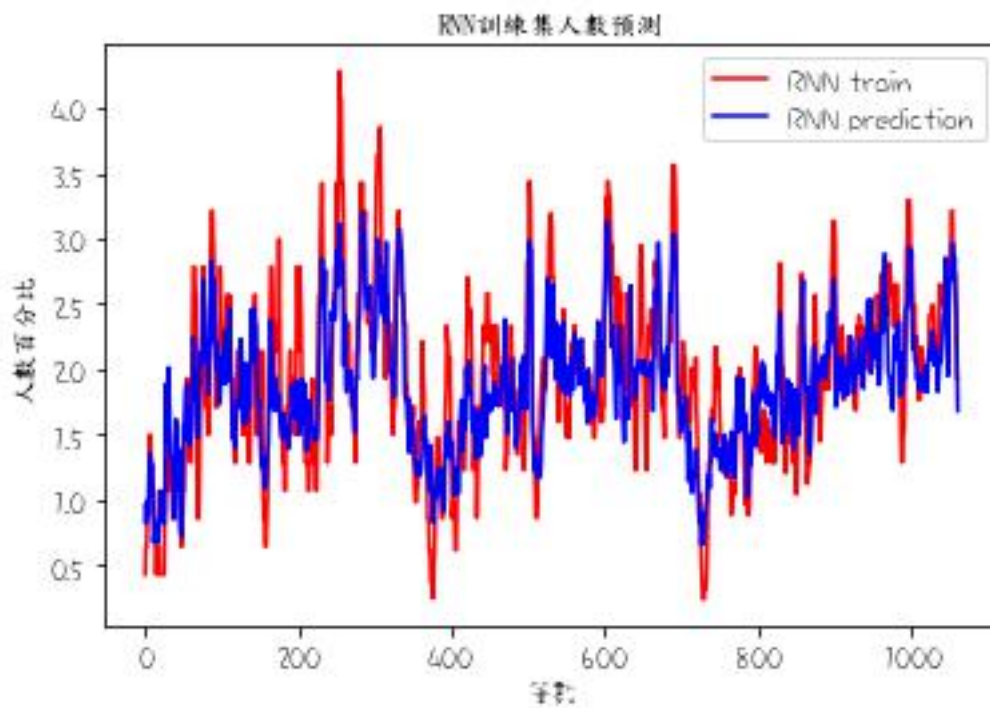


圖 47 RNN 時間長度 14 神經元個數 48、24 訓練集

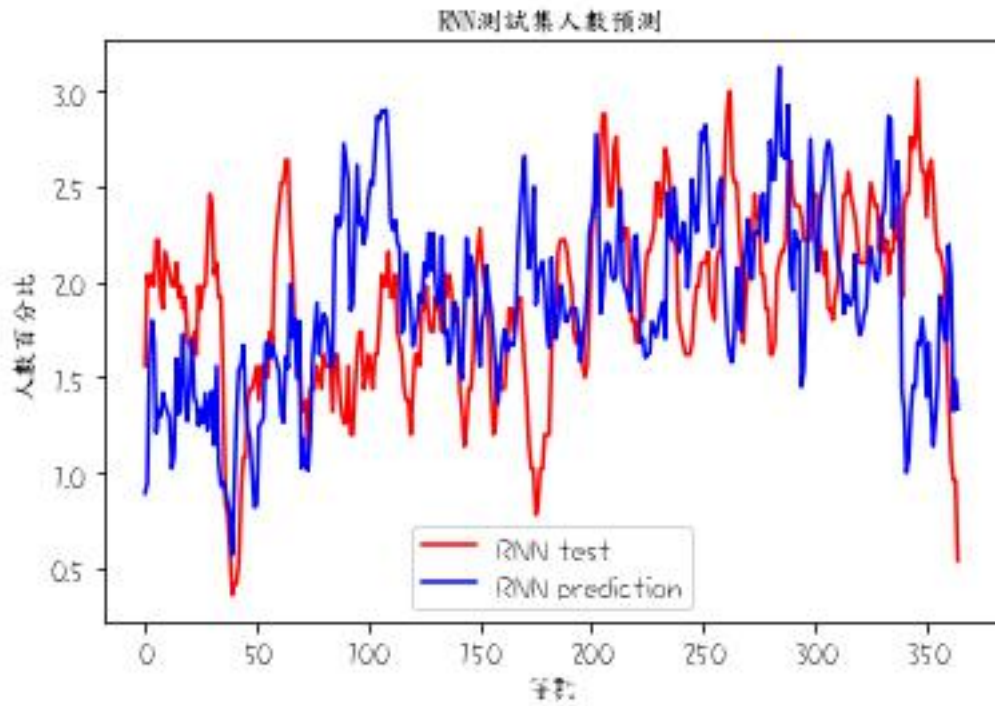


圖 48 RNN 時間長度 14 神經元個數 48、24 測試集

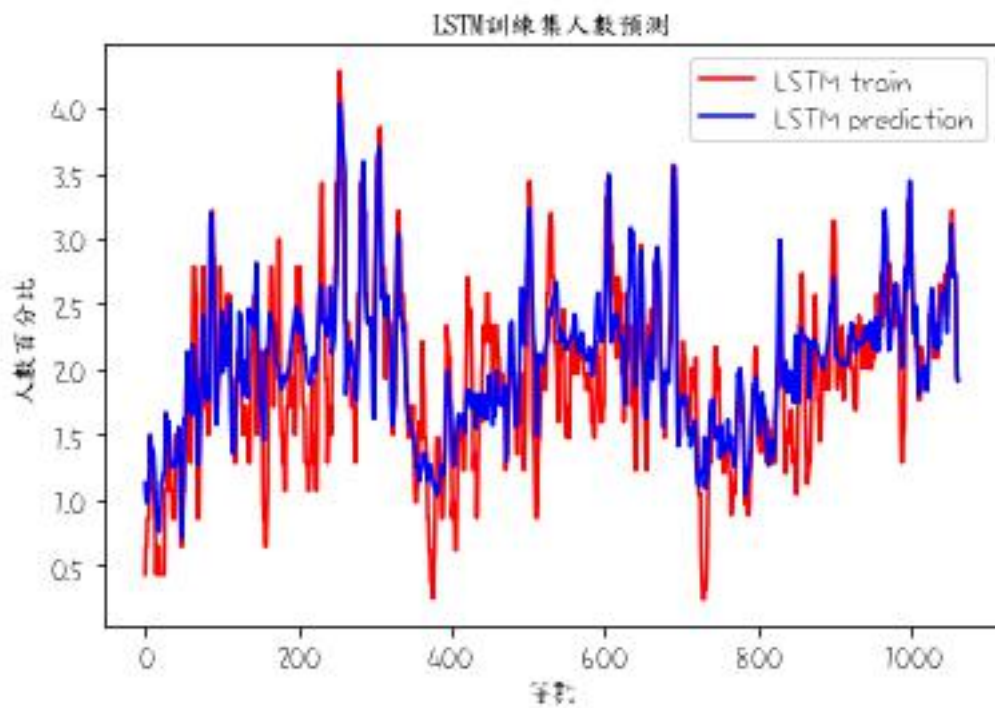


圖 49 LSTM 時間長度 14 神經元個數 48、24 訓練集

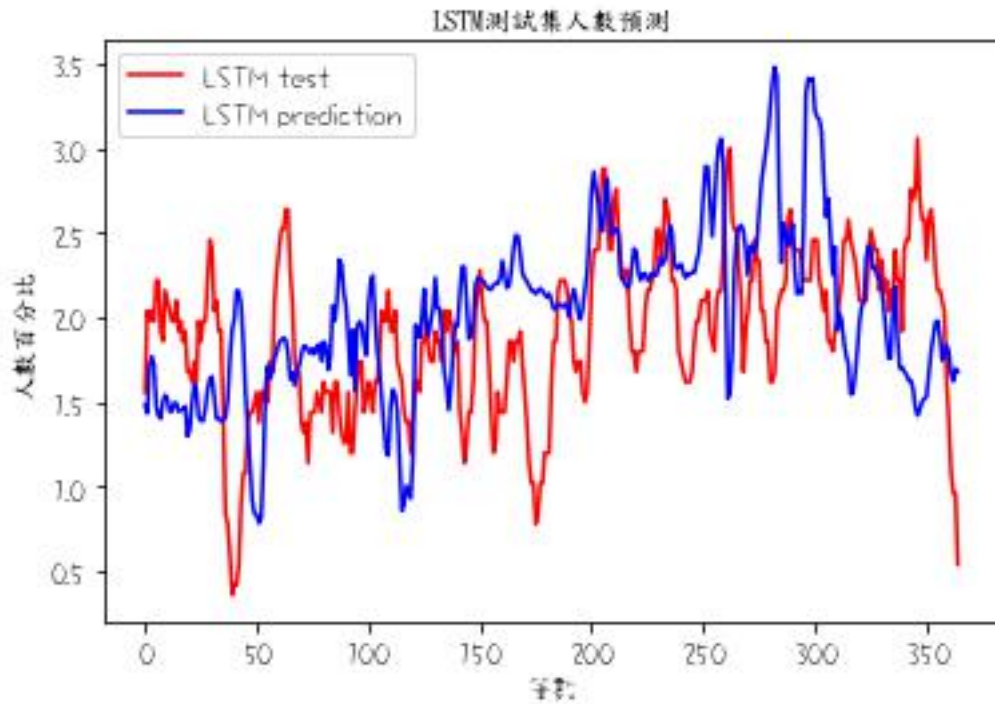


圖 50 LSTM 時間長度 14 神經元個數 48、24 測試集

(圖 51)為三種模型對訓練集與測試集的評估，從測試集來看，LSTM 的誤差為最小，代表神經元個數 48、24 較適合使用 LSTM 的模型來實作。

```
Dense NN訓練集: [0.09591063112020493, 0.23755790293216705]
RNN訓練集: [0.15195369720458984, 0.3093748688697815]
LSTM訓練集: [0.16591906547546387, 0.3102037310600281]
Dense NN測試集: [0.4314872920513153, 0.51568204164505]
RNN測試集: [0.3709646463394165, 0.5005293488502502]
LSTM測試集: [0.3862825930118561, 0.5043532252311707]
```

圖 51 神經元個數 48、24 訓練集與測試集損失與誤差

4.4.4 神經元 64、32

神經元個數 neurons1 與 neurons2 分別為 64、32，繪製出 DNN、RNN、LSTM 三種模型的訓練集與測試集的預測結果，如(圖 52-57)所示。

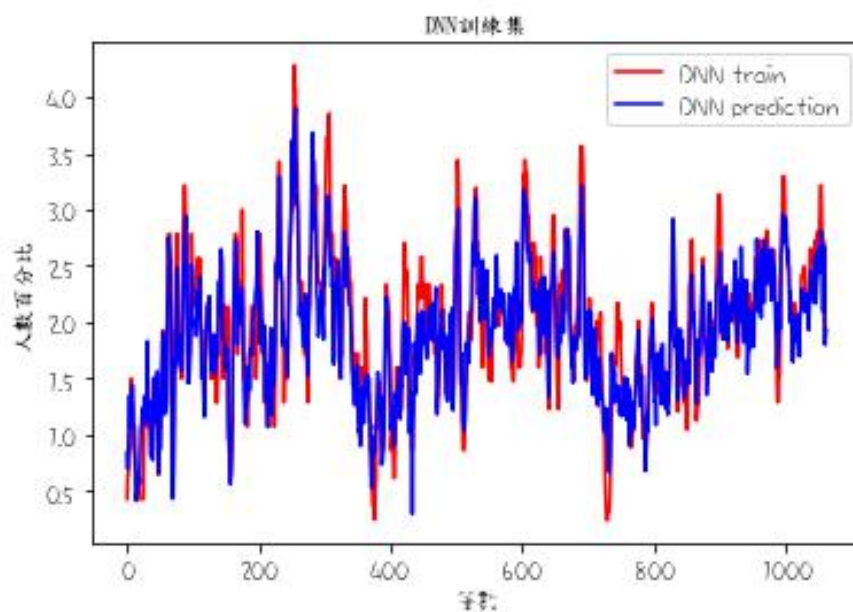


圖 52 DNN 時間長度 14 神經元個數 64、32 訓練集

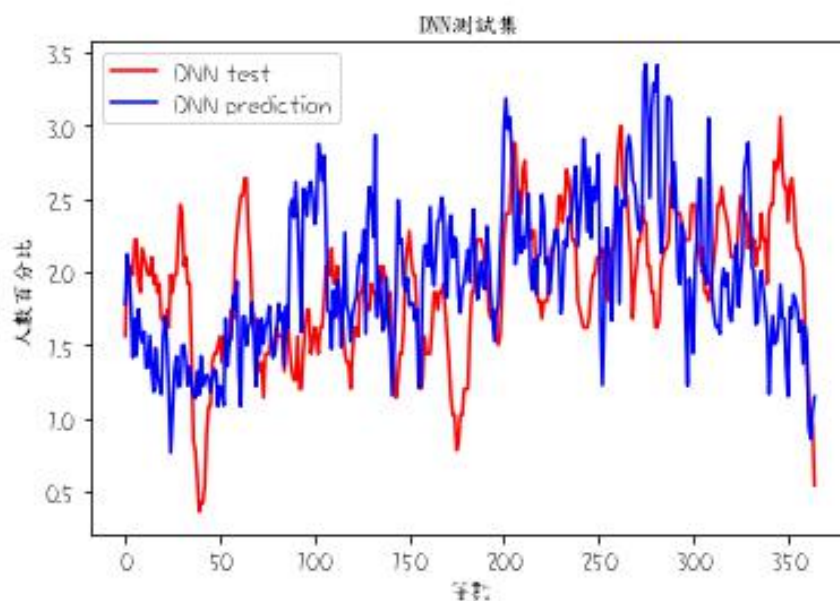


圖 53 DNN 時間長度 14 神經元個數 64、32 測試集

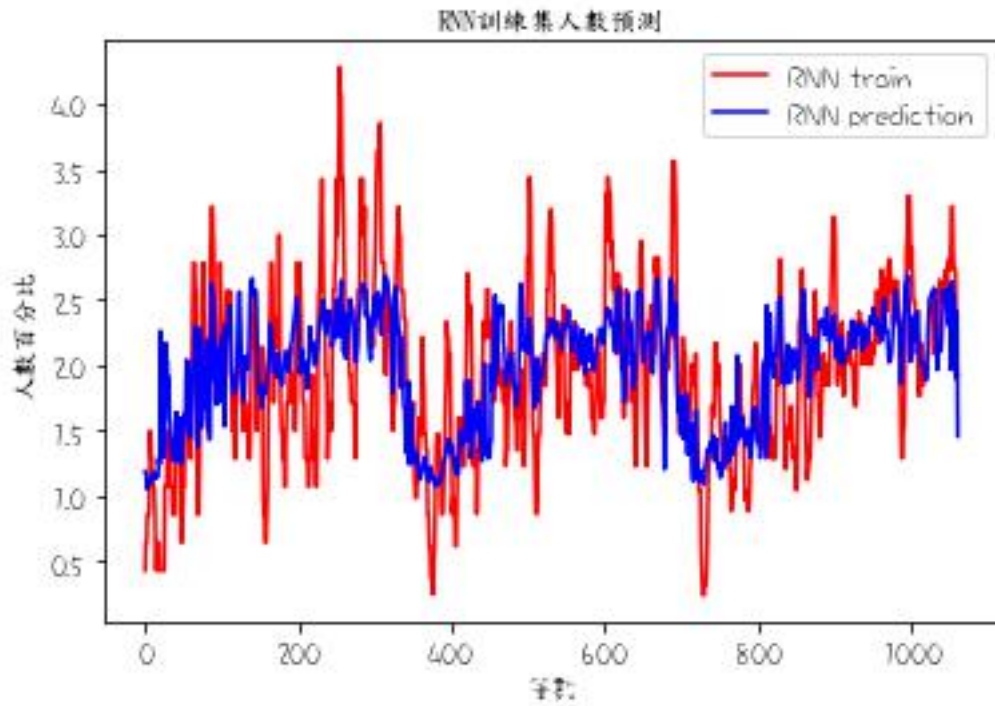


圖 54 RNN 時間長度 14 神經元個數 64、32 訓練集

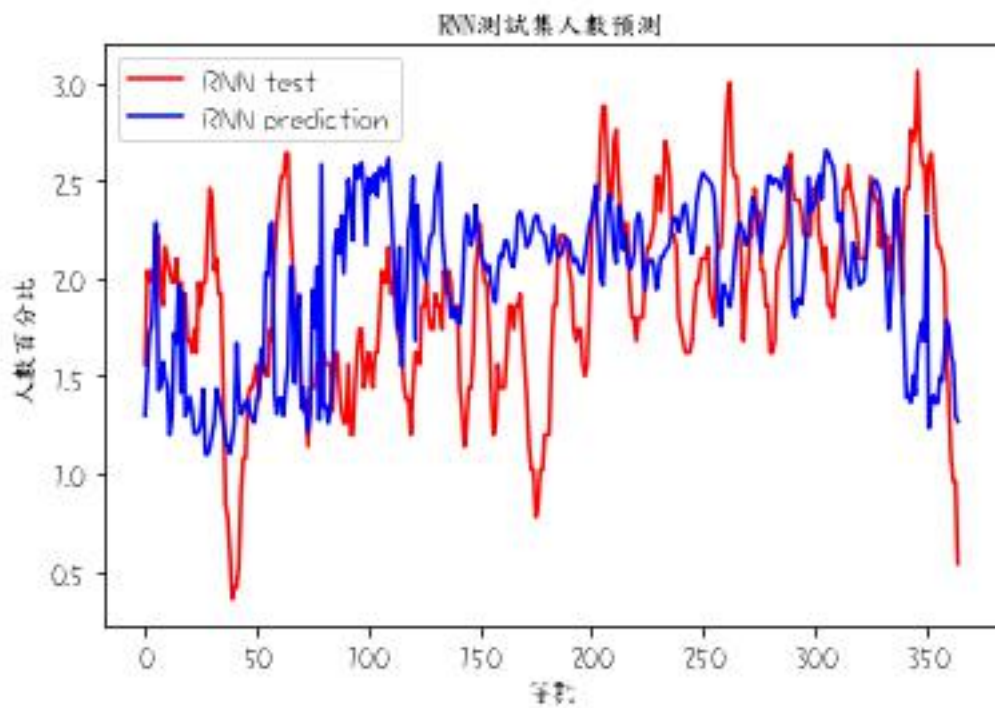


圖 55 RNN 時間長度 14 神經元個數 64、32 測試集

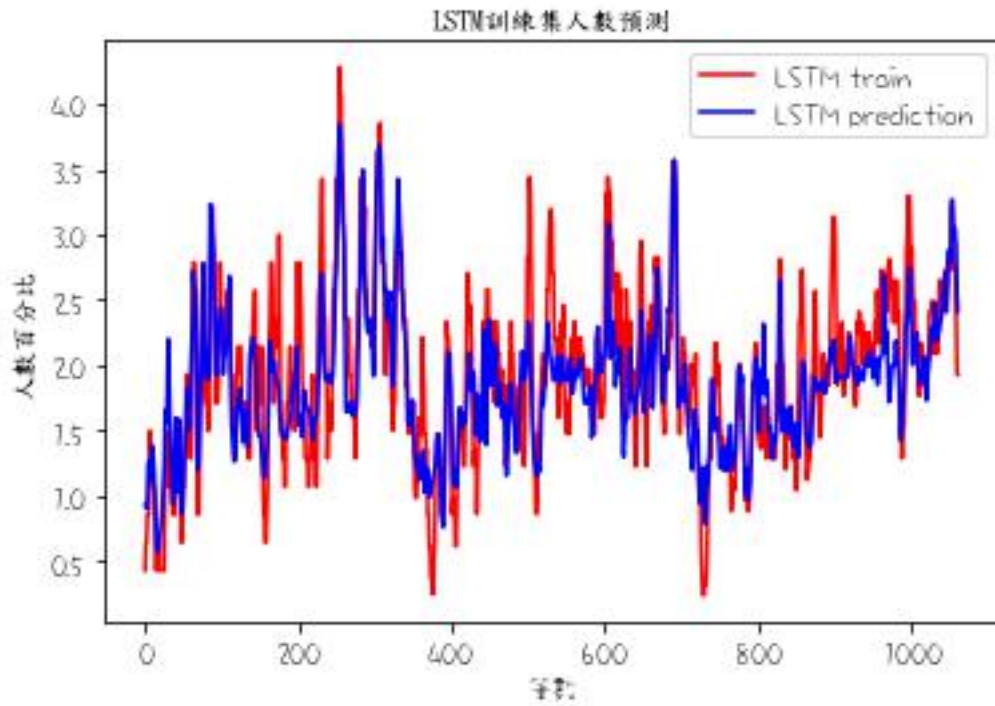


圖 56 LSTM 時間長度 14 神經元個數 64、32 訓練集

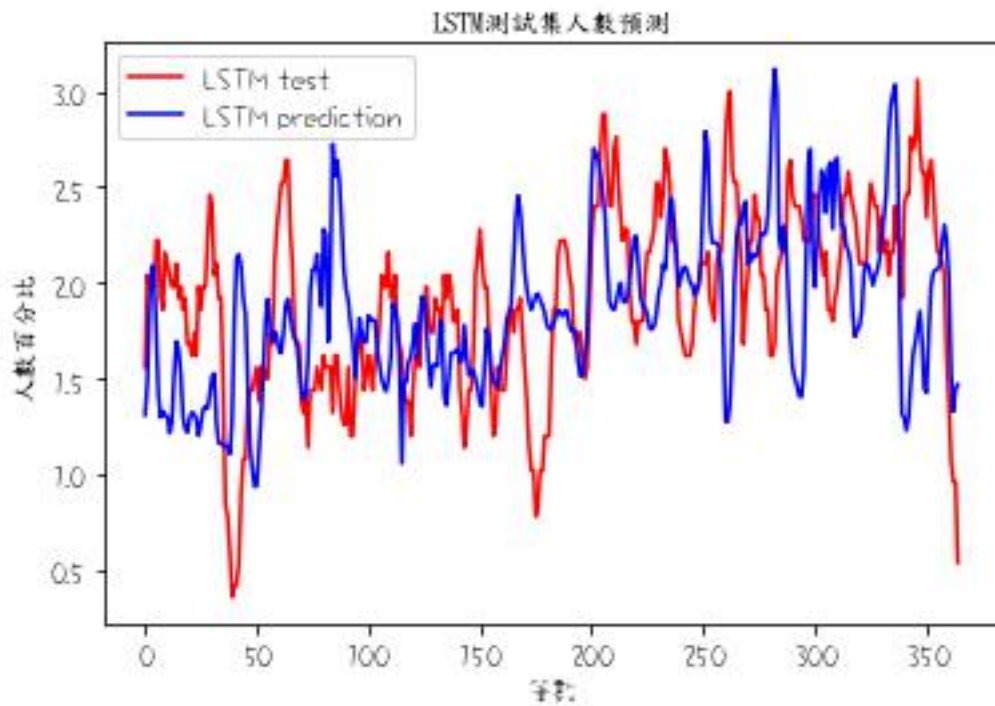


圖 57 LSTM 時間長度 14 神經元個數 64、32 測試集

(圖 58)為三種模型對訓練集與測試集的評估，從測試集來看，RNN 的誤差為最小，代表神經元個數 64、32 較適合使用 RNN 的模型來實作。

```
Dense NN訓練集: [0.07215217500925064, 0.2015809416770935]
RNN訓練集: [0.1514957696199417, 0.3023812174797058]
LSTM訓練集: [0.10219398140907288, 0.24855254590511322]
Dense NN測試集: [0.4124941825866699, 0.5120200514793396]
RNN測試集: [0.3264792859554291, 0.455534964799881]
LSTM測試集: [0.3926013708114624, 0.5064789056777954]
```

圖 58 神經元個數 64、32 訓練集與測試集損失與誤差

(表 4)為 4 組神經元測試集的比較，發現除了第四組誤差為 RNN 模型最小之外，第一組、第二組、第三組神經元個數之誤差皆為 LSTM 模型最小。就整體性而言，第二組神經元個數 32、16 的測試集誤差比其他三組的誤差來的小，故以神經元個數 32、16 來做後續時間長度 7、21 的三種模型比較。

表 4 4 組神經元測試集比較

	24 12	32 16	48 24	64 32
DNN	0.52648985	0.50375962	0.52325576	0.51202005
RNN	0.52456647	0.46534914	0.49336344	0.45553496
LSTM	0.49794409	0.4581795	0.48663133	0.5064789

4.4.5 不同時間長度的比較

以時間長度 14 為基準，去進行 4 組神經元個數測試集誤差之比較，發現神經元 32、16 測試集的誤差為最小，故採用神經元個數 32、16 來對時

間長度 7、21 進行預測。時間長度 7 三種模型的預測結果，如(圖 59-64)所示，時間長度 21 三種模型的預測結果，如(圖 65-70)所示。

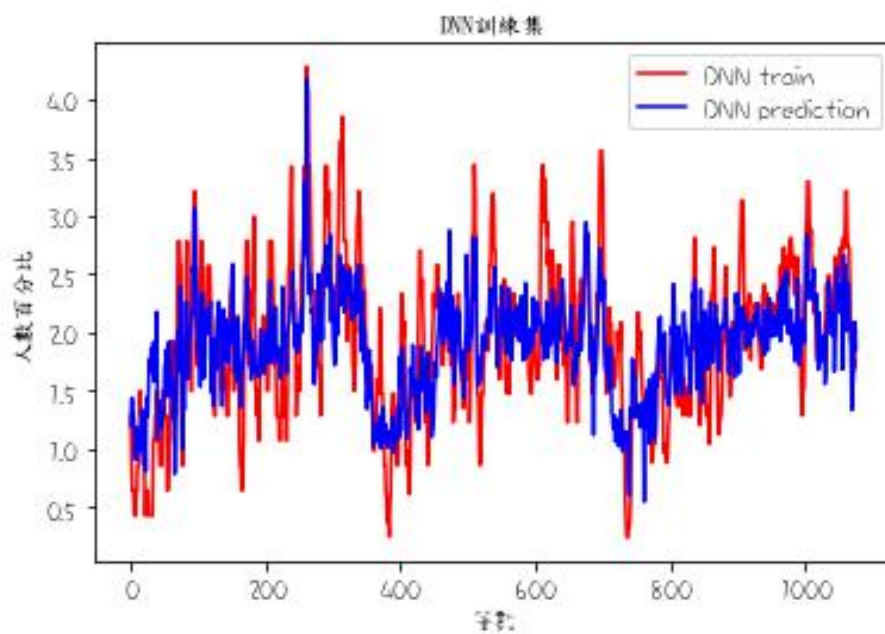


圖 59 DNN 時間長度 7 神經元個數 32、16 訓練集

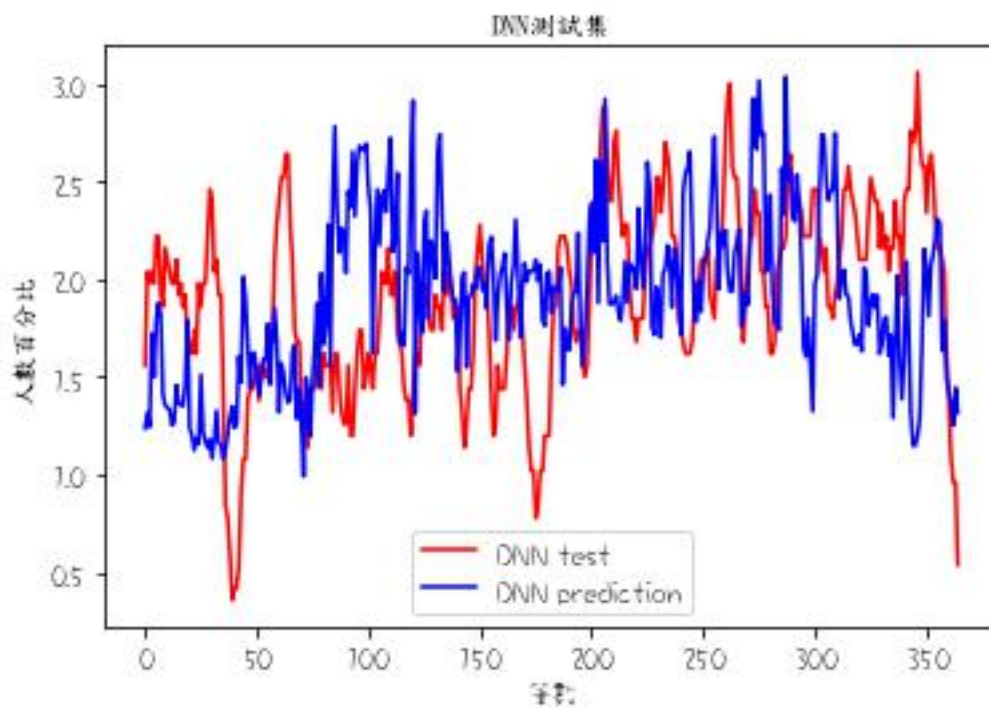


圖 60 DNN 時間長度 7 神經元個數 32、16 測試集

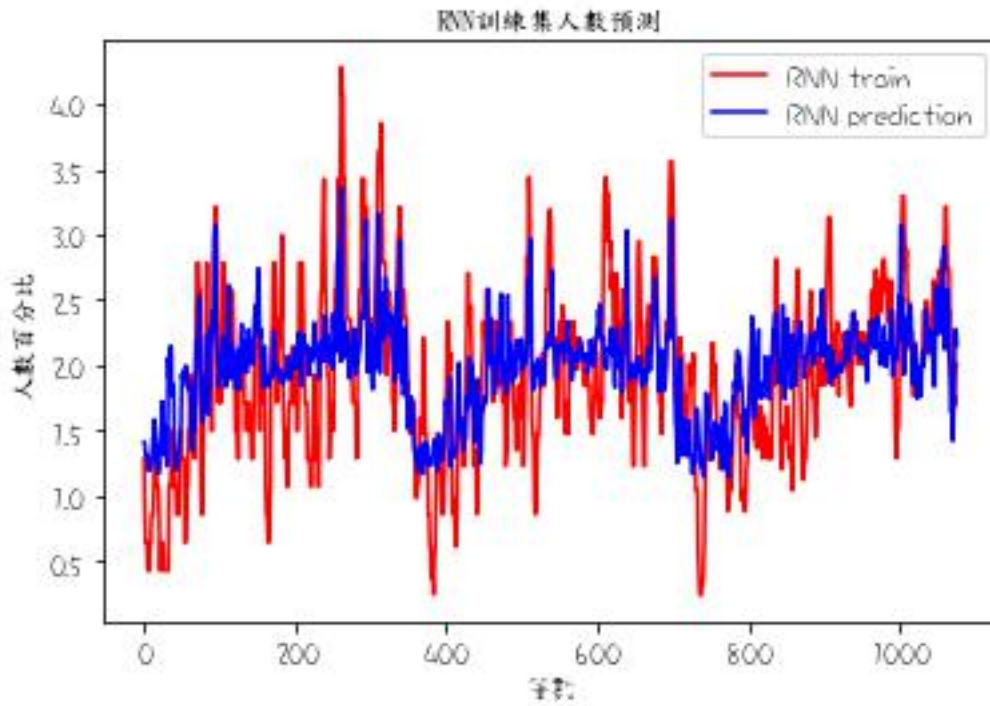


圖 61 RNN 時間長度 7 神經元個數 32、16 訓練集

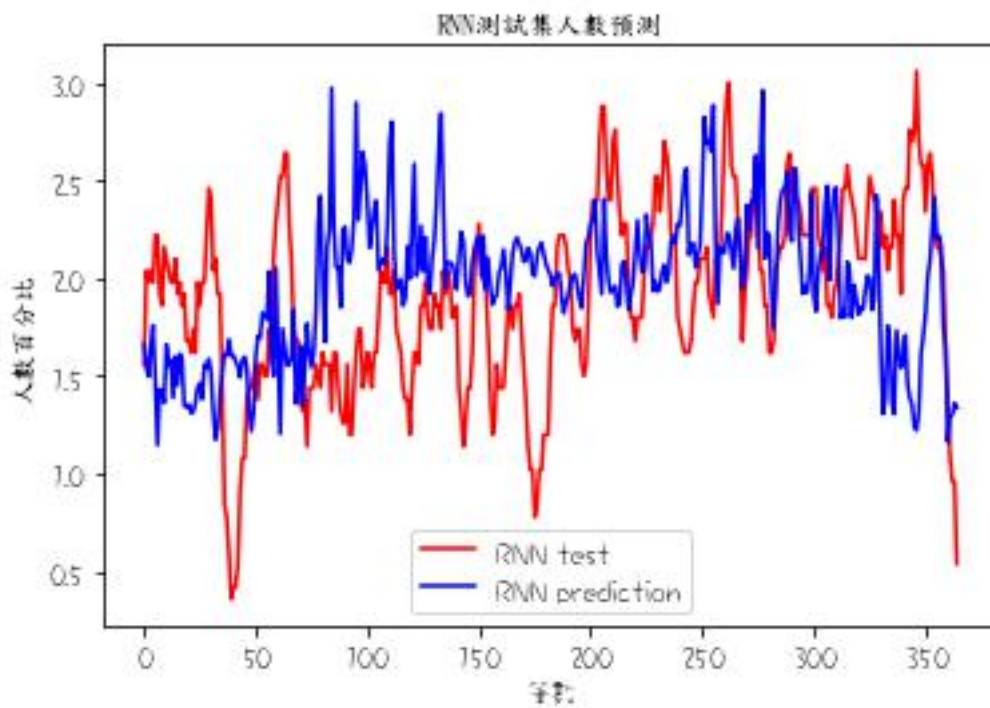


圖 62 RNN 時間長度 7 神經元個數 32、16 測試集

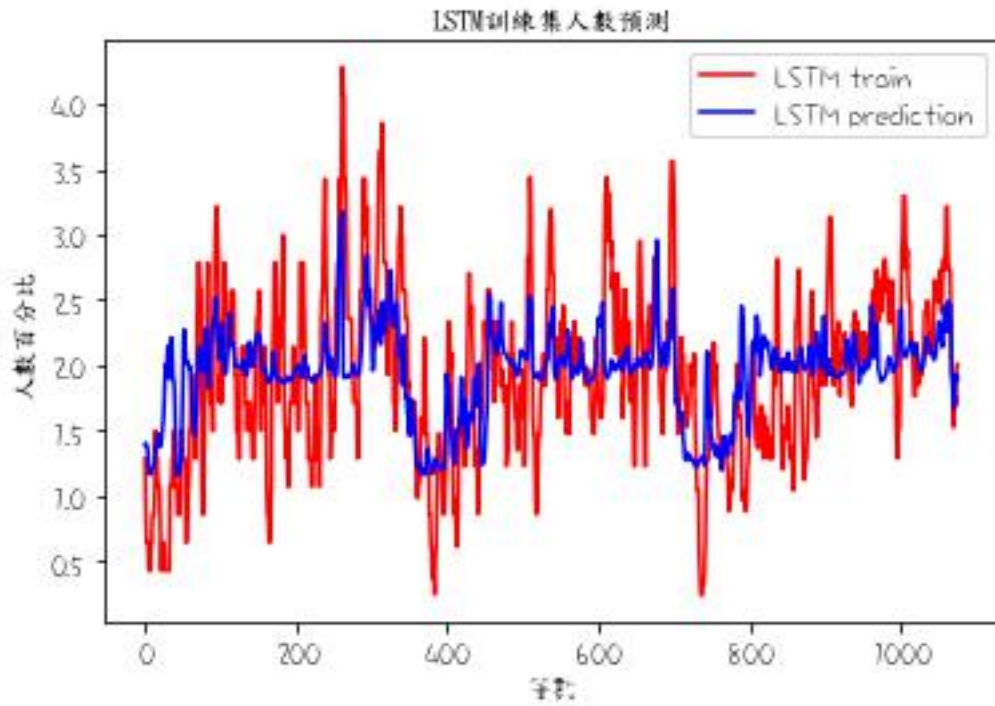


圖 63 LSTM 時間長度 7 神經元個數 32、16 訓練集

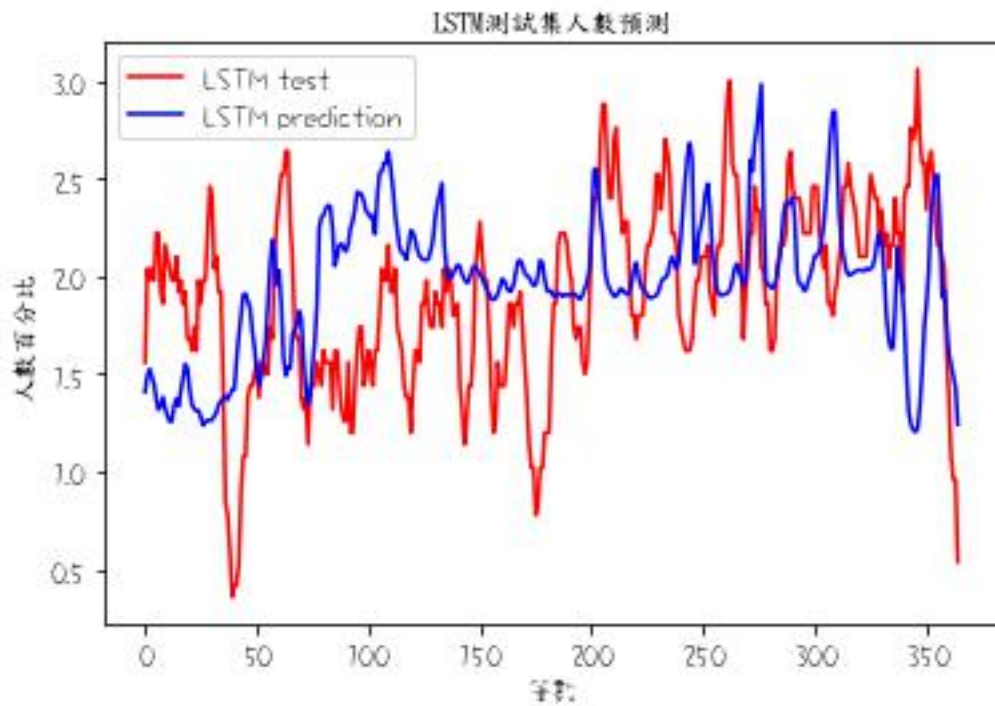


圖 64 LSTM 時間長度 7 神經元個數 32、16 測試集

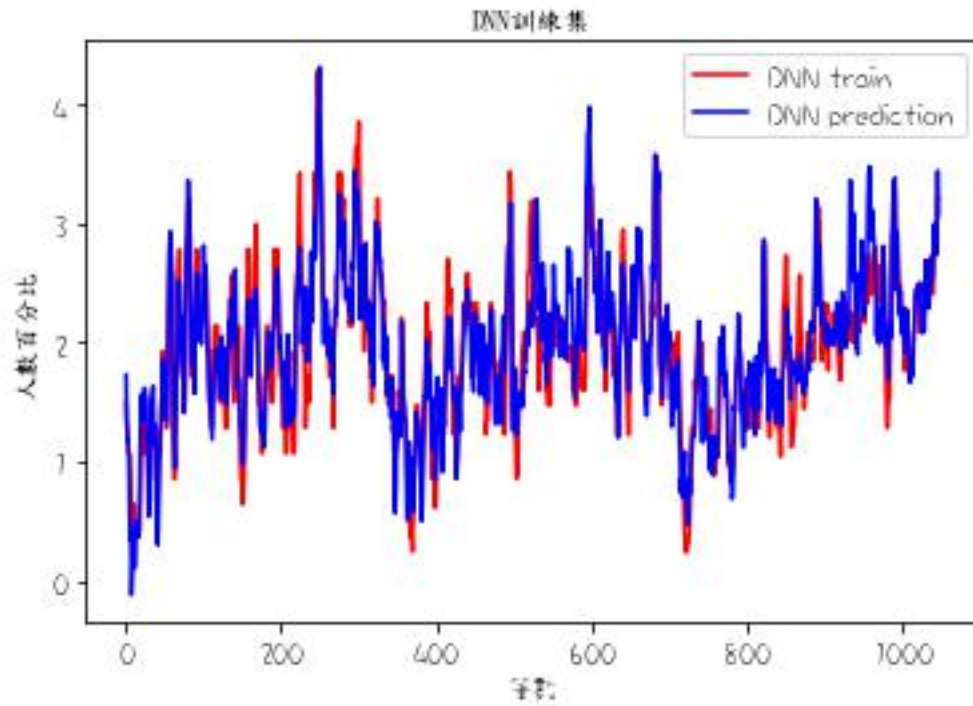


圖 65 DNN 時間長度 21 神經元 32、16 訓練集

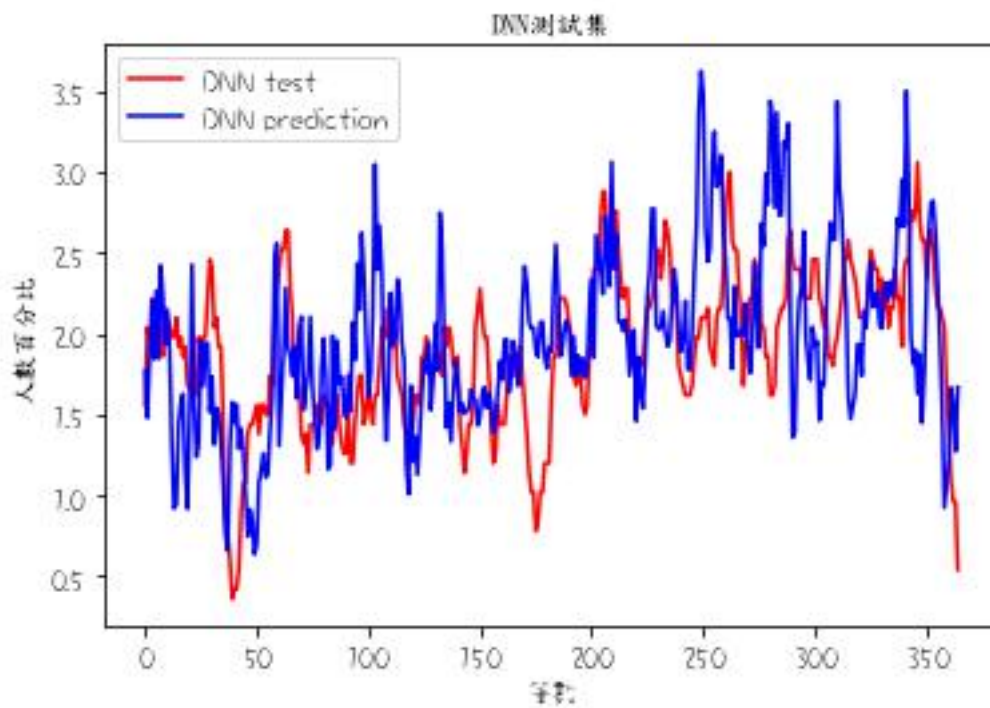


圖 66 DNN 時間長度 21 神經元 32、16 測試集

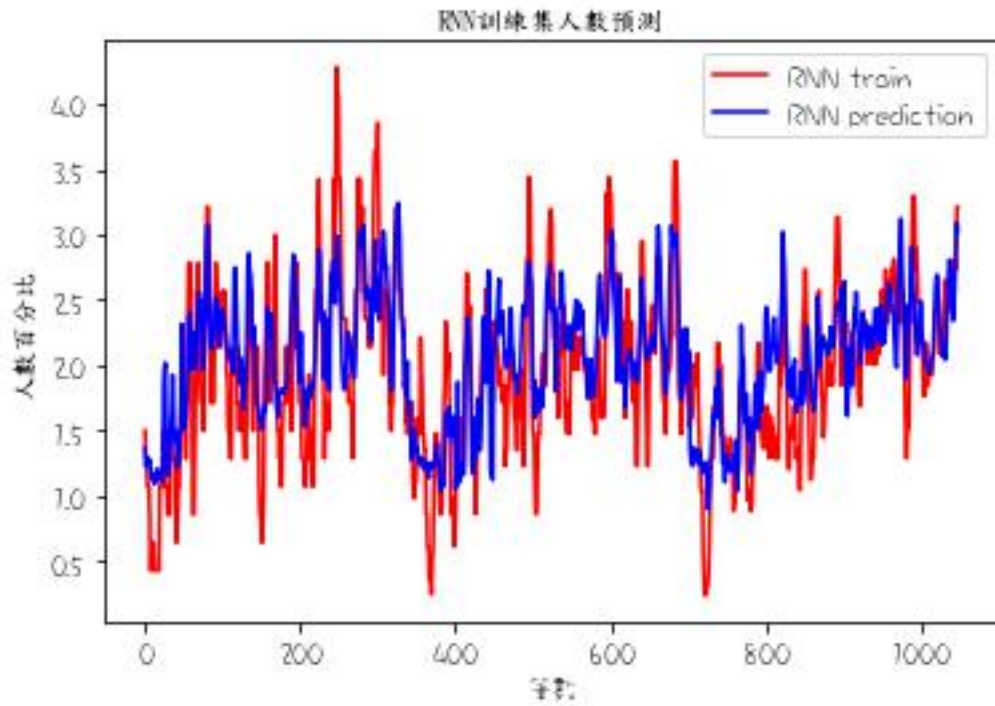


圖 67 RNN 時間長度 21 神經元 32、16 訓練集

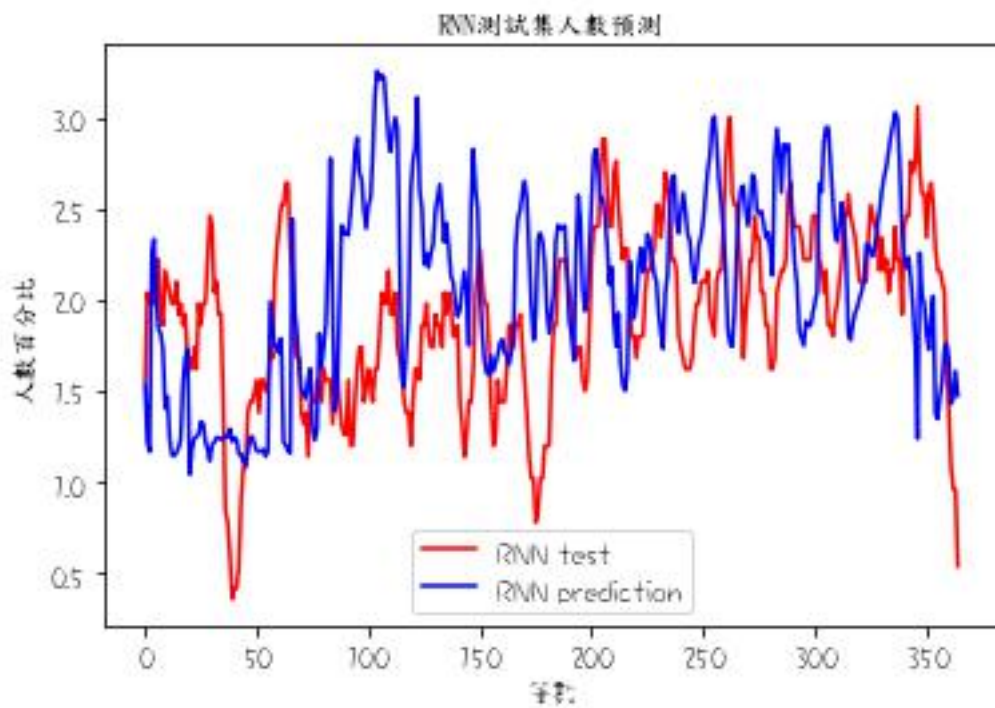


圖 68 RNN 時間長度 21 神經元 32、16 測試集

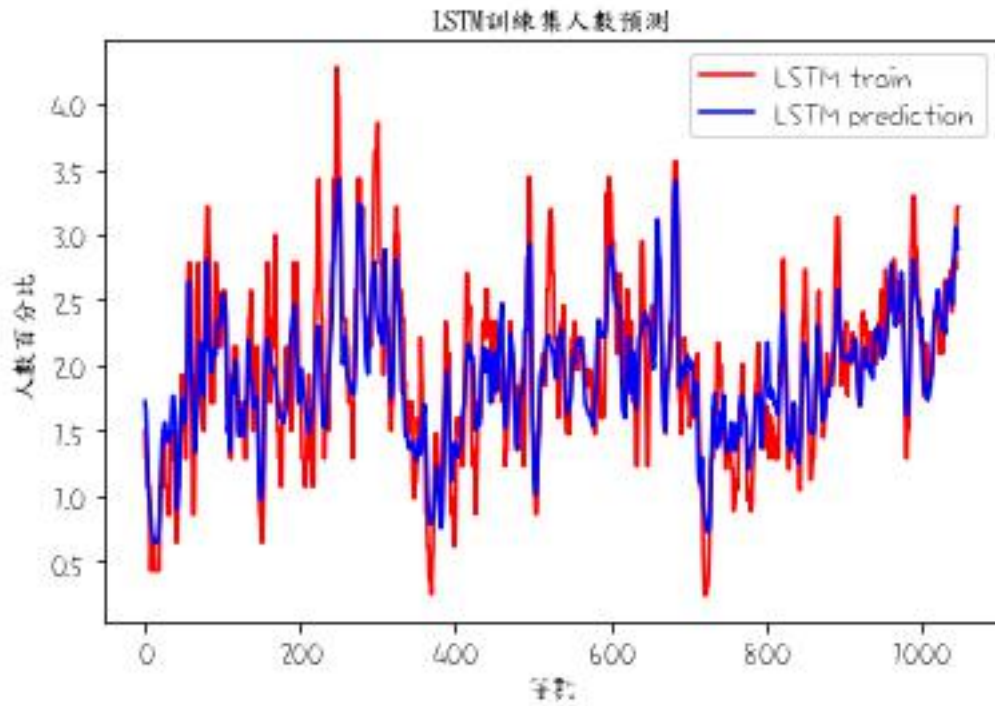


圖 69 LSTM 時間長度 21 神經元 32、16 訓練集

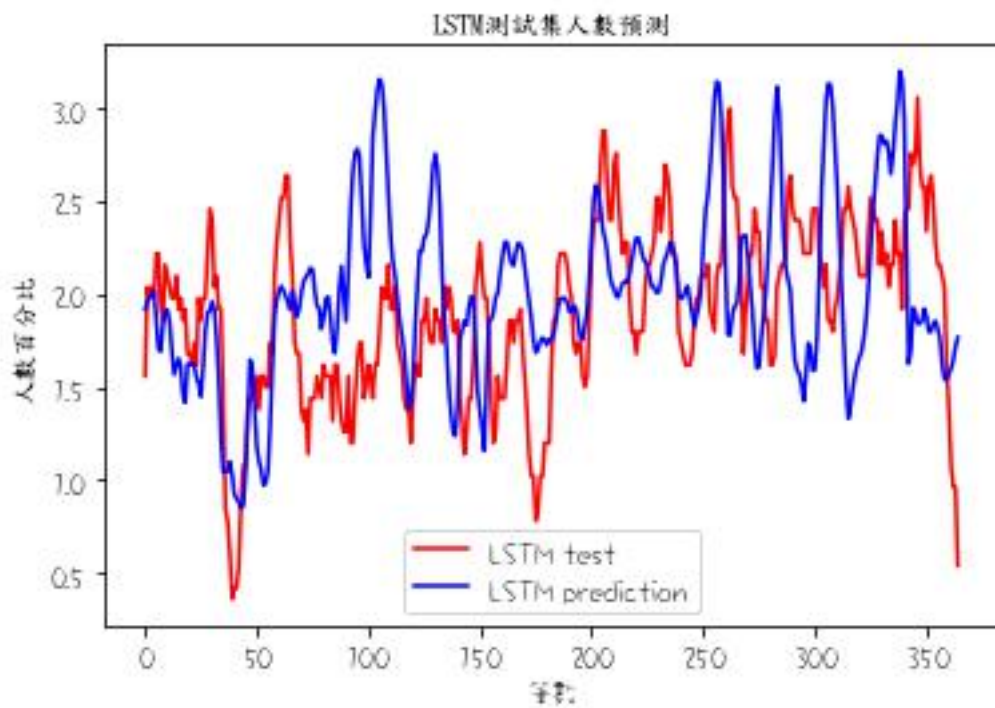


圖 70 LSTM 時間長度 21 神經元 32、16 測試集

(圖 71)為時間長度 7 三種模型對訓練集與測試集的評估，從測試集來看，LSTM 的誤差為最小，代表時間長度 7 較適合使用 LSTM 的模型來實作。

```
Dense NN訓練集: [0.2283506691455841, 0.37394651770591736]
RNN訓練集: [0.26249101758003235, 0.40159040689468384]
LSTM訓練集: [0.3074682056903839, 0.43613314628601074]
Dense NN測試集: [0.35964319109916687, 0.48997241258621216]
RNN測試集: [0.33497127890586853, 0.4699603021144867]
LSTM測試集: [0.324687123298645, 0.4674713909626007]
```

圖 71 時間長度 7 神經元 32、16 訓練集與測試集損失與誤差

(圖 72)為時間長度 21 三種模型對訓練集與測試集的評估，從測試集來看，LSTM 的誤差為最小，代表時間長度 21 較適合使用 LSTM 的模型來實作。

```
Dense NN訓練集: [0.09426014125347137, 0.23590752482414246]
RNN訓練集: [0.3141157627105713, 0.44785040616989136]
LSTM訓練集: [0.23698224127292633, 0.3816916048526764]
Dense NN測試集: [0.32483845949172974, 0.4527588188648224]
RNN測試集: [0.3370523452758789, 0.46650007367134094]
LSTM測試集: [0.28079915046691895, 0.44384798407554626]
```

圖 72 時間長度 21 神經元 32、16 訓練集與測試集損失與誤差

(表 5)為三種模型對時間長度 7、14、21 之比較，發現三個時間長度皆以 LSTM 模型最小，而時間長度 21 測試集的誤差相較於其他兩個時間長度小，故採用時間長度 21 之 LSTM 模型做後續分析。

表 5 三種模型不同時間長度的比較

	7	14	21
DNN	0.48997241	0.50375962	0.45275882
RNN	0.4699603	0.46534913	0.46650007
LSTM	0.46747139	0.4581795	0.44384798

4.4.6 類別準確度

經上面所述，類別準確度採用時間長度 21、神經元個數 32、16 之 LSTM 模型，(圖 73)使用 Kmeans 將人數百分比資料從數值型資料轉換成類別型資料，分類為 3 類後，做 one-hot encoding，將值變為 0 與 1。

做完 one-hot encoding 後，放入 TimeseriesGenerator() 中進行資料採樣，最後建立 LSTM 模型進行類別準確度之訓練及評估，如(圖 74)所示，建立模型時，由於已使用 Kmeans 方法金資料分為 3 類，故輸出層必須有 3 個神經元，並使用 softmax 啟動函數輸出機率值；編譯模型時，損失函數及評估函數與數值型預測模型不同，類別準確度的評估函數使用 acc 準確率，而損失函數使用最適合多元分類之 categorical_crossentropy。

```

# %% Kmeans
people_data = [[i] for i in people_data]
print(people_data)
# 分3群
from sklearn import cluster
k = 3

people_data2 = cluster.KMeans(n_clusters=k)
people_data2.fit(people_data)
print(people_data2.labels_)
people_data3 = np.choose(people_data2.labels_, [0,2,1]).astype(np.int64)

# 轉成one-hot encoding
from tensorflow.keras.utils import to_categorical

people_data3 = to_categorical(people_data3)
print(people_data3)

```

圖 73 進行 Kmean 分群

```

lstm_model = Sequential()
lstm_model.add(layers.LSTM(neurons1, input_shape=(length, 2)))
lstm_model.add(layers.Dropout(0.2))
lstm_model.add(layers.Dense(neurons2, activation='relu'))
lstm_model.add(layers.Dense(3, activation='softmax'))
lstm_model.summary()

lstm_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['acc'])

lstm_history = lstm_model.fit(train_gen,
                              epochs=100,
                              )
accuracy = lstm_model.evaluate(test_gen)
print("準確度 = ", accuracy[1])

```

圖 74 LSTM 準確率訓練及評估

(圖 75)為評估後之訓練集與測試集，準確度分別為 87.12%與 47.4%。透過混淆矩陣的表格顯示方式，讓模型評估出的結果，以視覺化的方式呈現，如(圖 76)所示，使用 `pd.crosstab` 建立混淆矩陣，輸入之 `temp2`(真實值)與 `prediction`(預測值)皆必須為 1 維陣列。(圖 77)為執行後顯示之混淆矩陣，數字 0、1、2 分別為人數百分比低、中、高，`predict` 與 `label` 分別為預測值及真實值，紅色框對角線是預測正確之筆數，其中真實是 2(高)，被正確預

測為 2(高)的筆數有 58 筆，佔真實值全部筆數 2(高)之機率為 78.38%最高，最不容易混淆；真實值是 1(中)，被正確預測為 1(中)的筆數有 78 筆，佔真實值全部筆數 1(中)之機率為 35.62%最低，最容易混淆。

```
175/175 [=====] - 0s 2ms/step - loss: 0.3208 - acc: 0.8712
訓練集準確度 = 0.8711832165718079
61/61 [=====] - 0s 2ms/step - loss: 1.7399 - acc: 0.4740
測試集準確度 = 0.4739725887775421
```

圖 75 LSTM 雙因子準確率

```
import pandas as pd
result = pd.crosstab(temp2, prediction,
                    rownames=['label'], colnames=['predict'])
result.to_html('confusion_matrixs2.html')
```

圖 76 混淆矩陣

predict	0	1	2
label			
0	37	26	9
1	59	78	82
2	8	8	58

圖 77 混淆矩陣顯示結果

4.5 單因子預測比較

針對平均氣壓與平均氣溫分別做數值的預測及類別的評估。程式與雙因子大致相同，但需要在 LSTM 模型輸入變數的 feature 特徵值進行調整，原來是雙因子，所以特徵值為 2，而這裡只需為 1。

4.5.1 平均氣壓

(圖 78-79)為平均氣壓的訓練集與測試集的預測曲線圖，(圖 80)為平均氣壓訓練集與測試集的損失與誤差，平均氣壓訓練集與測試集之誤差分別為 0.41717482 與 0.46849379。

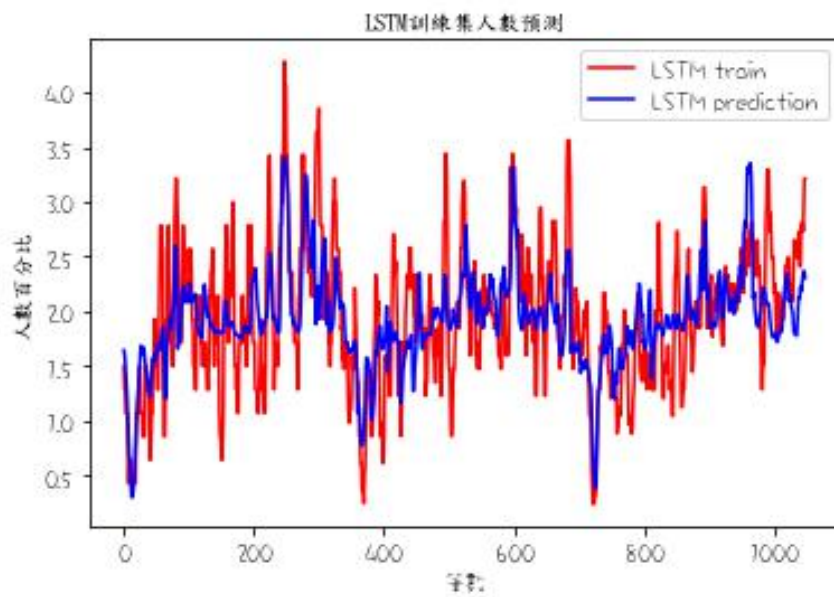


圖 78 平均氣壓訓練集預測

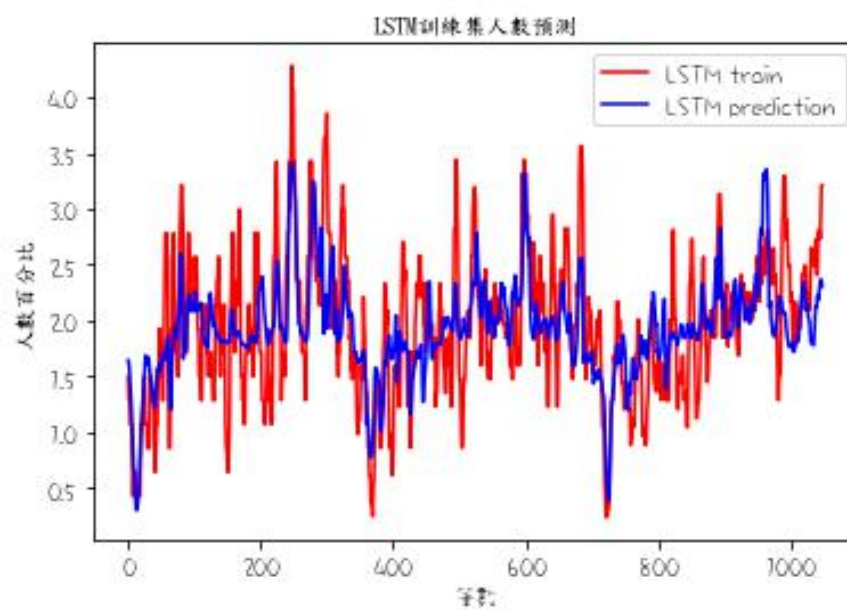


圖 79 平均氣壓測試集預測

LSTM訓練集: [0.29634225368499756, 0.4171748161315918]
LSTM測試集: [0.3586832880973816, 0.468493789434433]

圖 80 平均氣壓損失與誤差

4.5.2 平均氣溫

(圖 81-82)為平均氣壓的訓練集與測試集的預測曲線圖，(圖 83)為平均氣壓訓練集與測試集的損失與誤差，平均氣壓訓練集與測試集之誤差分別為 0.36479756 與 0.43714377。

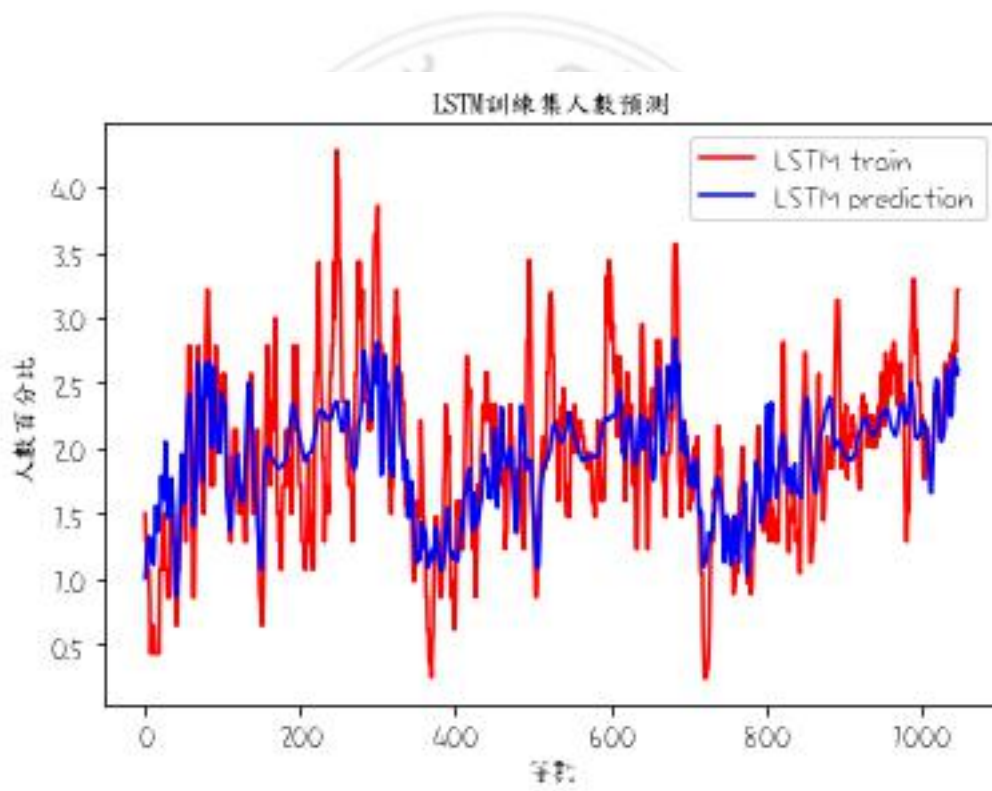


圖 81 平均氣溫訓練集預測

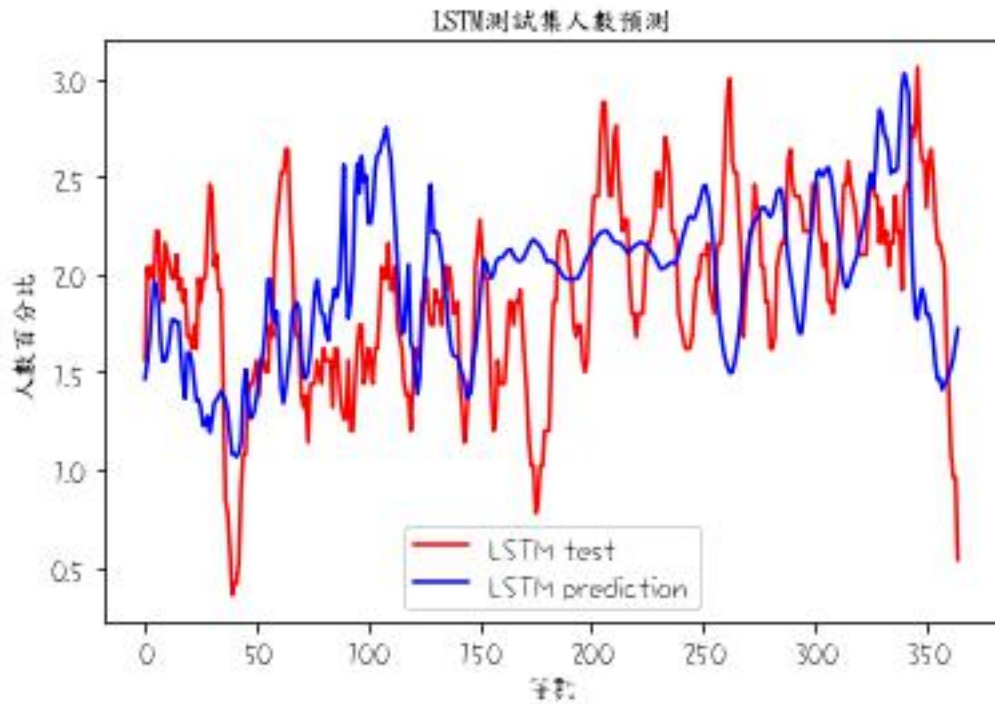


圖 82 平均氣溫測試集預測

LSTM訓練集: [0.219695046544075, 0.36479756236076355]
 LSTM測試集: [0.2854629158973694, 0.4371437728404999]

圖 83 平均氣溫損失與誤差

4.5.3 平均氣壓與平均氣溫比較

針對平均氣壓與平均氣溫的誤差進行比較，發現不論是訓練集或是測試集，平均氣溫都比平均氣壓低，代表平均氣溫比平均氣壓較適合使用時間長度 21、神經元個數 32、16 的 LSTM 模型進行實作，如(表 6)。

表 6 平均氣壓與平均氣溫誤差比較表

	訓練集	測試集
平均氣壓	0.41717482	0.46849379
平均氣溫	0.36479756	0.43714377

4.5.4 單因子類別準確度

對平均氣壓與平均氣溫進行訓練集與測試集準確度的評估，平均氣壓訓練集與測試集的準確度分別為 73.47%與 43.84%；平均氣溫訓練集與測試集的準確度分別為 66.13%與 51.78%，如(圖 84-85)所示。

```
175/175 [=====] - 0s 2ms/step - loss: 0.5894 - acc: 0.7347
訓練集準確度 = 0.7347328066825867
61/61 [=====] - 0s 2ms/step - loss: 1.2931 - acc: 0.4384
測試集準確度 = 0.4383561611175537
```

圖 84 平均氣壓準確度

```
175/175 [=====] - 0s 2ms/step - loss: 0.7399 - acc: 0.6613
訓練集準確度 = 0.6612595319747925
61/61 [=====] - 0s 2ms/step - loss: 1.1060 - acc: 0.5178
測試集準確度 = 0.517808198928833
```

圖 85 平均氣溫準確度

(圖 86)為平均氣壓執行後顯示之混淆矩陣，其中真實是 0(低)，被正確預測為 0(低)的筆數有 37 筆，佔真實值全部筆數 0(低)之機率為 51.39%最高，最不容易混淆；真實值是 2(高)，被正確預測為 2(高)的筆數有 25 筆，佔真實值全部筆數 2(高)之機率為 33.79%最低，最容易混淆。

predict	0	1	2
label			
0	37	27	8
1	63	98	58
2	25	24	25

圖 86 平均氣壓混淆矩陣

(圖 87)為平均氣溫執行後顯示之混淆矩陣，其中真實是 1(中)，被正確預測為 1(中)的筆數有 137 筆，佔真實值全部筆數 1(中)之機率為 62.56%最高，最不容易混淆；真實值是 2(高)，被正確預測為 2(高)的筆數有 19 筆，佔真實值全部筆數 2(高)之機率為 25.68%最低，最容易混淆。

predict	0	1	2
label			
0	33	37	2
1	21	137	61
2	7	48	19

圖 87 平均氣溫混淆矩陣

(表 7)為平均氣壓與平均氣溫準確度的比較，發現雖然平均氣壓的訓練集準確度 73.47%高於平均氣溫 66.13%，但學習過後經測驗之測試集 43.84%卻低於平均氣溫 51.78%。

表 7 單因子類別比較

	訓練集	測試集
平均氣壓	73.47%	43.84%
平均氣溫	66.13%	51.78%

4.6 全因子預測比較

4.6.1 數值資料預測

針對全部的氣象因子做為樣本資料對肝火上炎的人數百分比進行預測，(圖 88-89)為訓練集與測試集的預測曲線圖以及，(圖 90)為訓練及與測試集之損失與誤差。

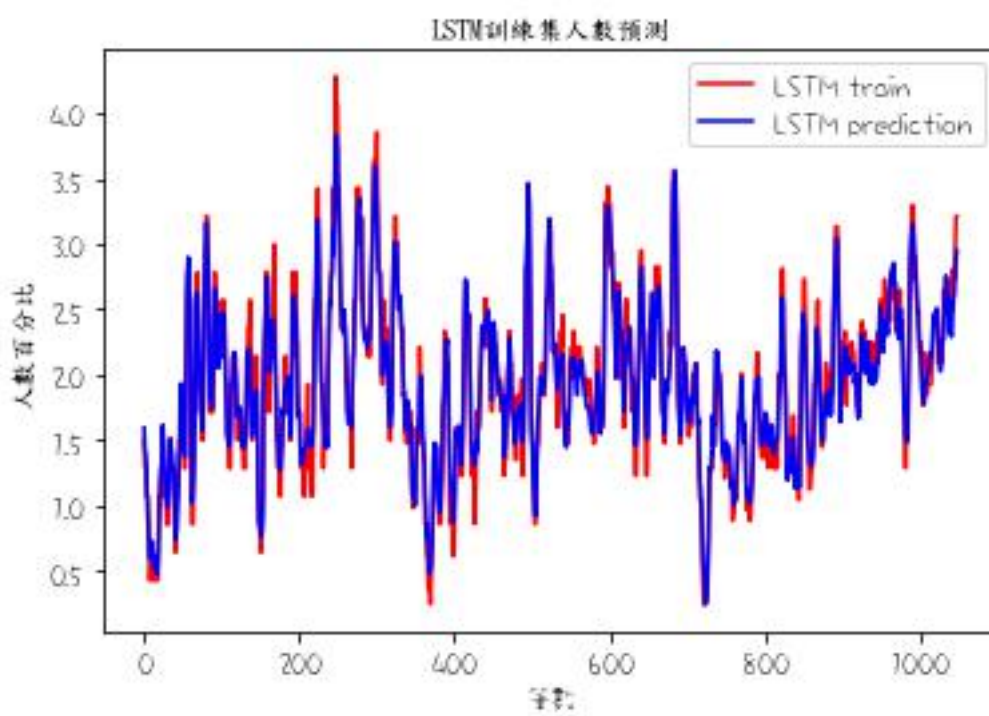


圖 88 全因子訓練集預測

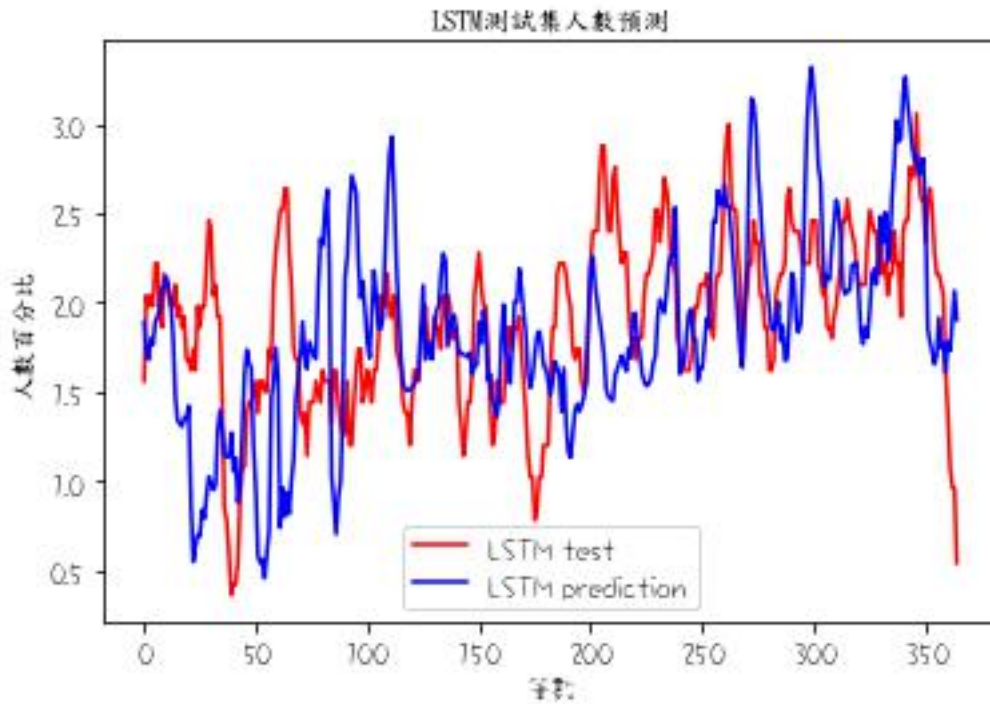


圖 89 全因子測試集預測

LSTM訓練集: [0.024189721792936325, 0.12139110267162323]
 LSTM測試集: [0.33537352085113525, 0.45780864357948303]

圖 90 全因子損失與誤差

4.6.2 類別準確度

將全部之氣象因子為輸入參數，對神經元個數 32、16，時間長度 21 之 LSTM 模型進行準確度評估，輸出後之訓練集為 97.33%、測試集為 47.95%(圖 91)。

```
175/175 [=====] - 0s 2ms/step - loss: 0.0841 - acc: 0.9733
訓練集準確度 = 0.9732824563980103
61/61 [=====] - 0s 2ms/step - loss: 2.2826 - acc: 0.4795
測試集準確度 = 0.4794520437717438
```

圖 91 全因子準確度

(圖 92)為平均氣壓執行後顯示之混淆矩陣，其中真實是 1(中)，被正確預測為 1(中)的筆數有 115 筆，佔真實值全部筆數 1(中)之機率為 52.51%最高，最不容易混淆；真實值是 2(高)，被正確預測為 2(高)的筆數有 22 筆，佔真實值全部筆數 2(高)之機率為 29.73%最低，最容易混淆。

predict	0	1	2
label			
0	38	30	4
1	61	115	43
2	12	40	22

圖 92 全因子混淆矩

4.7 單因子、雙因子、全因子預測比較

針對單因子、雙因子與全因子進行數值與類別的比較。在人數證候比例數值誤差中，訓練集最小為全因子 0.1213911，最大為單因子之平均氣壓 0.41717482，經過學習後測驗之測試集最小為單因子之平均氣溫 0.43714377，最大為單因子之平均氣壓 0.46849379。

在人數證候比例類別準確度中，訓練集最大為全因子 97.33%，最小為單因子之平均氣溫 66.13%，經過學習後測驗之測試集最大為單因子之平均氣溫 51.78%，最小為單因子之平均氣壓 43.84%。

平均氣溫在測試集數值誤差中是最小的，而在測試集類別準確度上是最高的，代表單因子平均氣溫較適合神經元個數 32、16，時間長度 21 之 LSTM 模型來實作。

在全因子的類別準確度上，訓練集準確度高達 97.33%，但經過學習後測驗之測試集卻只有 47.95%，代表其中可能具有雜訊、樣本資料量不足等問題。

表 8 單、雙、全因子比較表

		數值誤差		類別準確度	
		訓練集	測試集	訓練集	測試集
單 因 子	平 均 氣 壓	0.41717482	0.46849379	73.47%	43.84%
	平 均 氣 溫	0.36479756	0.43714377	66.13%	51.78%
雙因子		0.30485681	0.4581795	87.12%	47.4%
全因子		0.1213911	0.45780864	97.33%	47.95%

第五章 結論

鑑於近幾十年來氣候逐漸異常，導致各行各業進行策略及決定方針時，開始顧慮氣象因素，進而發展出氣象醫學、氣象經濟等與天氣相關之事物。本研究整合中醫證候時序資料與氣象因子時序資料，透過 Spyder 整合開發環境，使用 python 語言建立深度學習模型。以肝火上炎為主要目標，透過不同神經元個數、時間長度、資料型態與天氣因子數量，找出適合之深度學習模型。本章將針對研究所總結出研究結果、研究限制、未來發展與建議三部分。

5.1 研究結果

本研究建立 DNN、RNN、LSTM 模型，使用過去 7、14、21 天之氣象因子資料，預測未來 7 天後的肝火上炎人數百分比。透過不同神經元個數、時間長度、資料型態與天氣因子數量得出以下的結果：

1. 時間長度以過去 14 天為基準，天氣因子以平均氣壓與平均氣溫雙因子作為樣本資料，透過 4 組不同神經元個數之比較，得出神經元個數 32、16 之人數證候比例測試集之誤差，在其他 3 組神經元個數中最小。
2. 選用神經元個數 32、16 與另外時間長度 7、21 進行 DNN、RNN、LSTM 三種模型之比較，得出在神經元個數 32、16 中，時間長度 21 之 LSTM 模型之人數證候比例測試集誤差最小。
3. 選用神經元個數 32、16、時間長度 21 之 LSTM 模型，與另外兩種天氣因子數量單因子(平均氣壓、平均氣溫)、全因子進行人數證候比例之誤差及證候人數分類之準確度比較，得出單因子之平均氣溫在

人數證候比例測試集誤差最小，訓練集則為全因子最小，而在證候人數分類之準確度中，訓練集之準確度以全因子最大，而測試集之準確度以單因子之平均氣溫最大。

5.2 研究限制

調整神經網絡結構的過程中，增加神經元個數並未如預期的使預測表現更佳，反而使模型之誤差值增加，本研究推測很可能是因為時間序列資料之樣本數不足，提供更多之神經元反而增加更多的雜訊，造成更差之效果。

5.3 未來發展與建議

因目前從嘉義大林慈濟醫院中醫的門診資料取得的資料量只有 2010 年到 2013 年，而冀望後續的研究團隊能取得更多年份的資料，讓模型中的神經元有更多的資訊來進行預測。

本研究的 stride 參數為 1，每日都會取得目標資料，時間間隔太短，變動的幅度不會有明顯的變動，因此，本研究建議可將 stride 參數調大，使取得目標資料的時間間隔拉長，可能對準確度上有良好的提升。

本研究建議 RNN 與 LSTM 兩種模型可以與 1D 池化之 CNN 結合，因 RNN 與 LSTM 在處理長時間序列資料之運算時間較長，而 1D 池化之 CNN 可以作為預先處理步驟，在 RNN 與 LSTM 之前使用，能縮短序列資料及萃取資料特徵。

參考文獻

中文文獻

1. 賴昂廷、林益卿、楊鈺雯、吳美鳳 (2012)。氣候變遷與人類健康。彰化基督教醫院，家庭醫學科，內科學誌。
2. 萬擎、謝邦昌、廖佩珊(2016)。氣候及環境因子對重大心血管不良事件預測及相關因素探討。Journal of Data Analysis，11 卷 4 期。
3. 陳宗仁(2019)。氣象因子與癌症病患中醫實證的分析(碩士學位論文)。國立中正大學，嘉義。
4. 張慈文(2017)。中醫證治學。國防醫學院。
5. 楊孝兵、蔣峰、孫穎慧(2014)。氣象因素與類風濕關節炎病情活動度及中醫證候的關聯性。中醫雜誌。
6. 陳允傑(2019)。《TensorFlow + Keras：Python 深度學習應用實務》。出版地：旗標。
7. 黃晶一(2015)。氣象因素與成人社區獲得性肺炎中醫證型相關性研究(碩士學位論文)。廣州中醫藥大學，中國廣州。
8. Francois Chollet(2019)。《Deep learning 深度學習必讀：Keras 大神帶你用 Python 實作》。出版地：旗標。
9. Jiawei Han、Micheline Kamber、Jian Pei (2017)。《資料探勘》。出版地：高立圖書。
10. 胡建勳、劉堯欽(2008)。資料探勘於病患就診記錄之知識管理研究—以疾病與氣候關聯性為例。2008 知識社群與系統發展研討會。
11. 陳綺萱 (2019)。深度學習之專利分析研究。國立臺灣師範大學碩士學位論文，台北。
12. 張志聰(1670)。黃帝內經素問集注。出版地：西冷怡堂版。

13. 武月萍(2007)。氣象及氣象因素與中醫時間醫學。中華中醫藥學刊，45(3):p.20-23。
14. 楊威、劉銘福(2011)。隋唐時期疫病發生時令性的認識演進。中國中醫基礎醫學雜誌，17(3):p.253-254。
15. 羅賢君、張國強、林洪榮、張宇君、楊凱鈞、孫際恬、許鈞瓏(2018)。深度學習軟硬體加速器探索。電腦與通訊，174期，p.22-34。
16. 楊蕙菁(2004)。資訊也是商品統一超商砸四十億找出氣溫和銷售關係。商業週刊。
17. 張洪春、李得民、齊蕊涵(2019)。社區獲得性肺炎中醫證型的文獻研究。中醫雜誌 60(4):p.339-344。
18. 陳信賢(2015)。基於知識本體的中醫虛證辨證系統(碩士論文)。國立中正大學資訊工程研究所，嘉義。
19. 黃晶一、朱威、顏仕星、周紅(2018)。基于臨床大數據分析探討成人社區獲得性肺炎與氣象因素的相關性。中國中醫基礎醫學雜誌 24(6): p.782-785。
20. 史桂榮(2011)。試論二十四節氣與中醫學的關係。中醫藥導報 2011. 17(3):p.17-18。
21. 陳奕廷(2016)。機器學習與人工神經網路(二):深度學習(Deep Learning)。取自:<https://case.ntu.edu.tw/blog/?p=26340>。
22. 黃彥、周熙、陳貴川、劉德、田豐璋(2013)。重慶地區氣象因素對不同中醫証型骨痹患者發病的相關性研究。中國中醫急症 22(6):p.893-894。
23. 周熙、田豐璋、吳中朝、陳貴川、劉德(2013)。氣象因素對不同中醫証型骨痹患者發病相關性研究。成都中醫藥大學學報 36(4):p.38-42。

24. 劉清泉、呂小琴、張磊、丁雪霏、盧幼然、趙京霞、郭玉紅、胡晶、連博(2018)。2015年與2016年冬季北京地區流感樣病例與氣象因素關係的研究及流感確診病例中醫證候規律分析。27(4): p. 565-568.
25. 寺田學、辻真吾、鈴木たかのり、福島真太郎(2019)。用 Python 快速上手資料分析與機器學習(許郁文譯)。出版地：基峰。
26. 郝沛毅(2014)。資料探勘。出版地：高立圖書。



英文文獻

1. D'Amato, G., et al., Climate Change and Air Pollution: Effects on Respiratory Allergy. *Allergy Asthma Immunol Res*, 2016. 8(5): p.391-5.
2. D'Amato, M., et al., The impact of cold on the respiratory tract and its consequences to respiratory health. *Clin Transl Allergy*, 2018. 8: p.20.
3. Haogao Gu, et al., Meteorological Factors for Dengue Fever Control and Prevention in South China. *Int. J. Environ. Res. Public Health* 2016, 13, 867.
4. Yeh, Y.-H., et al., Seasonal variations of prescriptions for the major syndrome types and manifestations of upper respiratory tract infection in tradition Chinese medicine. *Complementary Therapies in Medicine*, 2016. 29: p.213-218.

