

國立中央大學

資訊管理研究所

碩士論文

類神經網路於投資策略的應用

Neural Network Applied in Investment Strategy

研究生：林奇鋒

指導教授：李俊賢 博士

中華民國 107 年 6 月

## 類神經網路於投資策略的應用

研究生：林奇鋒

指導教授：李俊賢博士

國立中央大學資訊管理學系碩士班

### 摘要

股票的波動是一種時間序列的資料。時間序列的預測是一個重要的研究議題，人工智慧計算模型目前正被廣泛使用於該議題，例如：類神經模糊系統等。本文提出複數型模糊類神經系統 (Complex neuro-fuzzy system) 並應用於多目標時間序列預測，此模型具有多組複數型態輸出，其中，每一組複數型態的輸出，其實部和虛部可分別針對兩個不同實數型態目標進行預測。有關特徵挑選，本研究採用多目標特徵挑選，篩選出針對所有目標有利的特徵，並以此作為模型輸入，以降低模型整體運算負擔及提高資料運用效率。在模型方面，由輸入層、複數模糊集合神經層 (Complex fuzzy sets layer)、前提式神經層 (Premise neural layer)、T-S 神經層 (Takagi-Sugeno neural layer) 及輸出層建構出多層式類神經網路。在參數學習方面，訓練模型時我們採用分治原則 (Divide-and-conquer principle)。複數模糊集合神經層的參數使用不同的演算法優化，像是粒子群演算法 (Particle swarm optimization, PSO)、人工蜂群演算法 (Artificial bee colony optimization, ABCO); T-S 神經層的參數使用遞迴式最小平方演算法 (Recursive least-squares estimation, RLSE) 進行優化; 其他的神經層沒有參數需要優化。在實驗方面，我們設計三個實驗檢驗模型的效能，將 PSO-RLSE 及 ABCO-RLSE 實驗結果結合投資策略，計算模型利潤互相比較也與不同的文獻方法比較。經由效能比較結果，本文提出多目標預測的研究方法表現出優秀效能。

**關鍵字：** 時間序列、複數模糊集、複數模糊類神經系統、人工蜂群演算法、粒子群演算法、遞迴最小平方演算法

## Neural Network Applied in Investment Strategy

Graduate student: Chi Feng Lin

Advisor: Dr. Chunshien Li

Department of Information Management, National Central University, Taiwan

### Abstract

Stock fluctuations are time series data. The prediction of time series is an important research topic. Artificial intelligence models are currently being widely used in this topic, such as neuro-fuzzy systems. This paper proposes a complex neuro-fuzzy system and applies it to multi-target time series prediction. This model has multiple complex-valued outputs, every output can have real and imaginary parts for two different real-valued targets, respectively. With regard to feature selection, this study uses multi-target feature selection to filter out features that are beneficial to all targets and use this as the model inputs to reduce the overall computational burden and improve data utilization efficiency. In terms of model, multi-layer neural network is constructed from input layer, Complex fuzzy set layer (CFS layer), Premise neural layer, Takagi-Sugeno neural layer (T-S neural layer), and output layer. For parameter learning, we use the divide-and-conquer principle when training the model. The parameters of the complex fuzzy set neural layer are optimized using different algorithm, such as particle swarm optimization (PSO), artificial bee colony optimization (ABCO); the parameters of the T-S neural layer are optimized using recursive least-squares estimation (RLSE), other neural layers have no parameters to optimize. In terms of experiments, we use three experiments to test the performance of the model. We combine investment strategy with PSO-RLSE and ABCO-RLSE experimental results, respectively, and calculate model profit to compare with each other and the different literature methods. Through the results of performance comparison, this paper presents a multi-target prediction method showing excellent performance.

**Keywords:** Time series; Complex fuzzy set; Complex neuro-fuzzy system; Artificial bee colony optimization; Particle swarm optimization; RLSE;

## 致謝

## 目錄

章節	頁次
類神經網路於投資策略的應用 .....	i
Neural Network Applied in Investment Strategy .....	ii
致謝 .....	iii
目錄 .....	iv
圖目錄 .....	vi
表目錄 .....	vii
第一章 緒論 .....	1
1.1 研究背景 .....	1
1.2 研究動機與目的 .....	2
1.3 研究方法概述 .....	4
1.4 論文架構 .....	4
第二章 文獻探討 .....	6
2.1 特徵選取 .....	6
2.2 複數模糊集合 .....	7
2.3 類神經網路 .....	8
2.4 複數類神經模糊系統 .....	8
第三章 系統設計與架構 .....	10
3.1 複數模糊集 .....	10
3.2 結構學習 .....	11
3.3 複數模糊類神經模型 .....	14
3.4 參數學習演算法 .....	17

3.4.1	粒子群演算法.....	17
3.4.2	人工蜂群演算法.....	18
3.4.3	遞迴最小平方演算法.....	19
3.5	投資策略 .....	21
第四章	實驗.....	23
4.1	實驗一：台灣股票加權指數單目標預測 .....	24
4.2	實驗二：台灣股票加權指數與恆生指數雙目標預測 .....	30
4.3	實驗三：台灣股票加權指數、道瓊工業指數、納斯達克和標準普爾 500 四目標預測.....	37
第五章	討論.....	46
第六章	結論與未來研究方向 .....	49
6.1	結論 .....	49
6.2	未來研究方向 .....	49
參考文獻	.....	51

## 圖目錄

圖	頁次
圖 1 模糊集合輸入空間(2 維) .....	12
圖 2 資料密度總和 .....	12
圖 3 複數類神經模糊系統模型 .....	15
圖 4 PSO 粒子更新位置示意圖 .....	18
圖 5 預測結果 .....	27
圖 6 模型預測誤差值 .....	28
圖 7 實驗一模型學習曲線 .....	29
圖 8 預測結果 .....	34
圖 9 模型預測誤差值 .....	35
圖 10 實驗二模型學習曲線 .....	36
圖 11 預測結果 .....	43
圖 12 模型預測誤差值 .....	44
圖 13 實驗三模型學習曲線 .....	45

## 表目錄

表	頁次
表 1 實驗一模型設定 .....	25
表 2 機器學習參數設定 .....	25
表 3 效能比較(TAIEX, 實驗一).....	26
表 4 十次重複實驗效能統計(TAIEX, 實驗一).....	26
表 5 模擬投資利潤表(實驗一) .....	26
表 6 模擬投資利潤比較表(實驗一) .....	26
表 7 實驗二模型設定 .....	31
表 8 機器學習參數設定 .....	31
表 9 效能比較(實驗二) .....	32
表 10 十次重複實驗效能統計(實驗二) .....	32
表 11 模擬投資利潤表(實驗二) .....	32
表 12 模擬投資利潤比較表(實驗二) .....	32
表 13 實驗三模型設定 .....	38
表 14 機器學習參數設定 .....	38
表 15 效能比較(實驗三) .....	39
表 16 十次重複實驗效能統計(實驗三) .....	39
表 17 模擬投資利潤表(實驗三) .....	39



# 第一章 緒論

## 1.1 研究背景

時間序列資料是以時間為依序所排列而成的資料序列，像是股市、匯率皆是一種時間序列資料。時間序列可視為依離散時間次序觀察的數據，可以數理方法或是統計模型進行分析，是現今社會進行常使用的方式。隨著網路快速發展，以及科技資訊的進步，導致資訊量的增加快速。利用機器學習或是資料探勘等人工智慧的方式，可以找出隱藏於資料背後所隱藏的關聯性或是規則，而這些知識是無法僅依靠人類智慧就可找出的。

時間序列資料的預測被廣泛的使用在各個領域上，已經有許多學術研究產出，如金融方面的匯率、股票漲跌等；除此之外，能源耗損、疾病預測等可運用在資源的分配，有助於國家或是機構做有效的資源配置。上述許多領域中，與我們自身最相關的為金融方面的預測，如何有效的運用金錢投資帶來利潤，是一個值得探討的議題。財務經濟方面的時間資料預測卻最為困難，因為當中包括許多影響因素，像是個別企業情況、全球情勢、總體經濟環境等。總總因素導致財務經濟的時間序列具有高度的變化。因此若是能夠有一預測效果良好的模型，配合有效的輸入資料以及正確的投資策略，便能協助投資者從中獲取利潤。

財經的預測中，類神經網路是最被廣泛使用的方法，從 2005 年開始有非常多針對真實世界中財經問題的研究產出，像是 Zhi-Bin et al. [44]使用了適應性神經模糊推理系統(Adaptive neuro-fuzzy inference system, ANFIS)與人工類神經網路(Artificial neural network, ANN)預測年度超額收益並比較效能，發現兩者預測效果都非常好；2006 年，Patel et al. [31]使用 ANFIS、模糊推理系統和 ANFIS 協助投資者做經濟上的決策；Yao [41]於 2007 年提出一個方法針對外幣做買賣策略，從預測匯率中獲利；Li et al. [25]學者使用群體智慧與自我組織的類神經模糊系統(Neuro-fuzzy systems, NFSs)預測人民幣與美金的匯率；除了匯率之外，股票預測

也是有許多的預測產出，如 Abbasi 於 2008 使用 ANFIS 配合四個獨立變數，交易量、本益比和每股盈餘預測股票收盤價[1]；於 2017，Koijen et al. [22]透過股票和債券預測彼此的數據；Pan et al. [30]透過預測股票的波動性，比較短期和長期的槓桿效應。

## 1.2 研究動機與目的

在股市裡面，有著幾個著名的理論，其一為 Kendall et al. [18]1953 年提出的隨機漫步理論 (random walk theory)，意即股價的變動是獨立的，其間並無規律性或週期性存在；另一延伸為有效市場假說 (efficient market hypothesis, EMH)[11]，是學者 Fama 於 1970 年所提出，認為買方和賣方的價錢是雙方願意的，且假設所有市場之參與者均可無償取得資訊，因此投資人所收集的資訊並不能使其獲得超額利潤。但另有一派的專家認為股票是可以預測的，因此不斷的研究，於 1990 年 Kimoto et al. [21]利用倒傳遞類神經網路 (Back propagation neural network, BP NN)，搭配兩種指標來預測日經指數的漲跌和買賣時機；1999 年 Yao et al. [40]採用多項技術指標以倒傳遞類神經網路來預測股票市場；Kim et al. [20]於 2000 年將類神經網路中加入基因演算法 (Genetic algorithm, GA)，不但改進了類神經網路的學習速度，且降低了特徵空間的複雜度，實驗結果相對於倒傳遞類神經網路顯得更加優秀，並發現非線性類神經網路預測能力較好；2012 年，Wei [38]提出了基於經驗模態分解 (Empirical Mode Decomposition, EMD)的混合式適應性類神經模糊推論系統 (Adaptive network-based fuzzy inference system, ANFIS)，並配合投資策略計算利潤，在虛擬投資中，有著不錯的獲利效果；同年，Li et al. [26]以差分整合移動平均自迴歸 (Autoregressive integrated moving average, ARIMA)模型結合類神經模糊系統 (Neuro-fuzzy system, NFSs)預測股票的收盤價，隔年，Li et al. [27]使用複數類神經模糊系統對雙目標同時預測。

從上述可得知，類神經模糊系統對財經方面的預測非常的有用，且一個經過良好訓練的模型預測效能會精準許多，因此本研究將會以類神經模糊系統為基準，結合其他人工智慧技術，並比較訓練前後對模型預測的影響。在進行模型訓練的過程中，經常發生預測偏差或是模型運算過久等問題。因此如何挑選有效的輸入資料以及模型整體的大小是大多研究的主要議題。模型在進行預測的過程中，參數需要經過學習以及最佳化的過程來增加效能表現，目前有許多機器學習演算法被使用於尋找參數最佳解，像是粒子群最佳化演算法 (Particle swarm optimization, PSO)[19]、人工蜂群演算法 (Artificial bee colony optimization, ABCO) [17]、基因演算法 (Genetic algorithm, GA) [10]等。目前文獻大多是以 Takagi-Sugeno 模糊系統建立架構 [36]，因此所需最佳化的參數較多，會讓 PSO 等演算法面臨求解的參數維度過大，使得效果不如預期，發生收斂過於緩慢或落入區域最佳解的問題。

本研究將使用混合的演算法優化參數，分別以 PSO 及 ABCO 結合遞迴最小平方誤差 (Recursive least-square estimation, RLSE)演算法 [16]，形成 PSO-RLSE [23] 和 ABCO-RLSE。以分治法使用不同的演算法訓練不同部分的參數，減低機器學習在尋求最佳解的過程中，落入區域最佳解的機率。如上述所提，除了股票目標本身，股票的波動是由諸多因素所形成，因此本實驗將從多組股票中進行特徵挑選，也同時一次預測多個目標，從大量的資料中萃取對每個目標都有效益的內容。為了使預測效果提升，模型的建置上應用了類神經網路系統結合 T-S (Takagi-Sugeno)的模糊規則型態組成類神經模糊模型。並且將類神經模糊模型的參數分為兩部分，分別為前鑑部參數 (If-part parameters)與後鑑部參數 (Then-part parameters)。

傳統上，在建立模糊集的設計多是以高斯函數建立模糊集(Fuzzy sets) [43]，資料輸入後可得到介於 0 到 1 的隸屬程度值。Ramot 在 2002 年提出了(Complex Fuzzy Sets, CFSs) [32]，將原本隸屬程度加上一個虛部值，將隸屬程度的值從傳統的一維區間投影到實數軸 x 與虛數軸 y 兩個維度上的單位圓盤，使其相較於原本的實數型隸屬函數，具有較豐富的資料量。

### 1.3 研究方法概述

本研究提出混合型的演算法，PSO-RLSE 以及 ABCO-RLSE，用以最佳化模型參數。模型中有兩部分參數，分別為前鑑部參數與後鑑部參數，PSO 與 ABCO 負責優化前鑑部參數，RLSE 則優化後鑑部參數，以降低各個演算法的搜尋維度。

在本研究中，首先會利用夏農資訊熵 [35] 的概念對資料分析，並做多目標特徵選取 (Multi-target feature selection) [28]，藉由計算原始資料對目標所提供的資訊量作為標準，並且加入冗餘資訊的計算，選出對目標有良好影響的資料作為輸入至模型的依據，可以降低資料複雜度與計算成本。之後透過 PSO-RLSE 與 ABCO-RLSE 演算法，調整前鑑部參數及更新後鑑部參數。最後將結果與投資策略結合，計算出模型帶來的利潤。

### 1.4 論文架構

本論文一共分為六章，第一章為緒論，介紹本研究的背景、動機及研究之目的、概述本研究所要使用的方法及流程。第二章為研究理論方法的介紹及概述，包括過去特徵選取的一些方式，除此之外，將會介紹模糊集的起源以及複數模糊集的概念。接著介紹類神經網路的理論，以及相關的研究，最後提及被廣泛使用的類神經模糊系統。第三章為系統設計與架構，概述如何進行多目標特徵挑選，以及機器學習決定模型大小的方法；複數類神經模糊網路中各神經層的介紹也會在此章節敘述。接著會詳細說明參數學習的演算法，包括 PSO、ABCO、RLSE，概述演算法彼此混合後進行參數學習與訓練的過程，最後會提及投資策略的操作及概念。第四章是實驗及結果，本研究用了 3 個實驗進行時間序列預測之研究。實驗一利用台股加權指數進行單一目標的預測，實驗二是同時進行台股加權指數與上海恆生指數的預測，實驗三則是一次針對四個目標預測，分別是台股加權指

數、道瓊工業指數、納斯達克指數以及標準普爾 500。以上實驗皆會做兩種比較，其一為不同混合演算法的效能比較，包含訓練前後的差異，其比較標準包含配合投資策略後的利潤；除此之外也會與過去文獻所提出的方法進行效能比較，以證實本研究方法的優劣性。第五章為針對實驗的結果進行討論。第六章則是本研究的結論以及未來的研究方向。

## 第二章 文獻探討

### 2.1 特徵選取

資料計算模型是透過機器學習或資料探勘後所建構，像是分類、回歸、分群等。資料前處理的目的有兩個，其一是降低資料集合的大小，使得分析更具效果；其二是可針對特定的資料集合選出較適當的資料。在現今龐大的資料中，前者對整個分析方法來說是很重要的，因此特徵挑選在資料探勘、網路異常檢測、文字分類、基因分析等許多領域中扮演著不可或缺的角色。特徵集合的縮減是根據特徵的關聯性，以及他們對目標所造成的冗餘性 [42]。特徵挑選的方法可以被分類為數種 [14]，最常見的就是過濾法(Filter approach) [12]、包裝法(Wrapper approach) [2]、嵌入法(Embedded) [12]、混合法(Hybrid approach) [15]。以下為各方法的簡略說明。

過濾法(Filter method)是由 Guyon [12]於 2003 年所提出，此方法首先對資料集進行特徵選擇，接著將選擇出來的特徵子集合進行相關性評估並評分，最後設定門檻值個數，藉此選擇特徵，特徵選擇的過程與後續的模型無關，其優點是比包裝法還要快速。於 1997 年，Blum 所提出的包裝法(Wrapper approach) [2]，與過濾法最大的差別在於特徵子集合的挑選評分標準，其評分標準是由模型的運算結果所決定，因此在挑選過程中，會不斷的使用到模型計算，運算時間長且計算複雜，對於某些特定的分類器有著較高的分類準確度。嵌入法(Embedded approach) [12]是將特徵挑選與模型的訓練一同進行，因此在模型訓練結束後也會完成特徵挑選，時間與模型運算介於上述兩種方法之間。混合法(Hybrid approach) [15]是把過濾法及包裝法的概念結合，透過過濾法挑選出適當的特徵子集合，接著將這些特徵子集合透過模型運算選出最終的特徵，此方法提高了過濾法的精準度，也降低了包裝法的運算時間，結合了兩者的優點。

如上述，特徵集合的縮減，是根據特徵的關聯性與冗餘性，故本文採用熵的

概念計算特徵彼此間提供的訊息量。熵一詞最早是由德國物理學家 Rudolph Clausius 於 1854 年提出 [8]，是一種對物理系統之無秩序或亂度的量度，當熵值越高代表混亂程度也越高。關於資訊的亂度，夏農提出了夏農資訊熵(Shannon information entropy)的理論[35]，若資訊的隨機性越高，則資訊熵值會越高，對於某一個隨機變數 $X$ ，資訊熵的定義如下。

$$H(X) = - \sum_{i=1}^n p(x_i) \log p(x_i) \quad (1)$$

其中， $H(X)$ 是隨機變數 $X$ 的資訊熵； $p(x_i)$ 則是事件 $x_i$ 的發生機率； $\log\left(\frac{1}{p_d(x)}\right)$ 則被視為 $p_d(x)$ 的資訊混亂度。由公式可以發現發生機率越大，資訊熵值越小。反之，若事件出現機率趨於平均，則熵值會趨近最大值。如前所敘，熵值越大代表所提供的資訊隨機性越高。本研究基於夏農資訊熵(Shannon information entropy)的理論 [35]，進行多目標特徵挑選[28]，此方法中考慮到資訊量以及冗餘資訊量，最後配合使用者設定篩選出數個特徵當作模型的輸入。

## 2.2 複數模糊集合

過去集合的概念只有「屬於」或「非屬於」，但此概念卻無法應用於真實世界中，好比溫度若大於 30 度時稱為「熱」，但 29.9 度與 30 度並無太大差異卻不屬於「熱」，此情況與人類的思維有所衝突，因此在 1965 年，Zadeh 學者提出了模糊集的概念[43]，藉由此概念，我們可以透過一歸屬函數計算出元素屬於集合的隸屬程度，隸屬程度則介於 0 到 1 之間，隸屬程度越高，則該元素屬於集合的程度越高。之後 Buckley 提出了模糊複數(Complex number)的概念 [3-5]，2002 年，Ramot et al. [32]更進一步提出了複數模糊集合(Complex fuzzy sets, CFSs)，意即歸屬程度可以透過函式得到一個複數型態的值，這使得歸屬程度可以呈現在一個半徑為 1 的複數單位圓盤中。這個概念使原本能表示的歸屬程度更加的豐富。

## 2.3 類神經網路

類神經網路，是一種模擬人類大腦的技術，人類的大腦包含著大量的神經元，這些神經元主要功用是處理資訊和記憶，神經元之間是由「突觸」所連接，而「突觸」的功用如同權重，控制著資訊的流通，於 1943 年，McCulloch et al. [29]學者使用一種稱為閾值邏輯(Threshold logic)的演算法模擬出神經元的概念，當一個神經元接收到資訊，會判斷是否要產生興奮反應的機制。而後心理學家 Hebb 提出了海伯理論(Hebb's law) [13]，意即突觸前的神經元與突觸後的神經元同時活化時，則這兩個神經元之間連結的強度會增加；此想法於 1956 年 Rochester et al. [33]將之轉化成數學模型創造了感知機，以模擬人類大腦，可以說是當今深度學習(Deep learning)的始祖，但相較來說感知機神經元較少，且傳遞訊號權重為 0 或 1，仍然無法處理異或問題(非線性問題)，且當時電腦沒有足夠能力運算。直到 1975 年 Werbos [39]提出反向傳播演算法 (Back propagation, BP)，這個演算法有效地解決了異或問題以及訓練多層神經網路的問題。日後有許多其他的人工智慧方法被提出，像是支援向量機(Support vector machine, SVM) [9]、貝氏分類器(Bayesian classifier) [34]等，直到近年來，因資訊科技快速的發展，電腦所能運算的速度倍增，在某些專長領域已可超越人類水平，神經網路才再度被大量研究。

## 2.4 複數類神經模糊系統

在過去有許多研究方法被提出如 ARIMA、模糊理論、神經網路運算等。其中，最常被提出的就是類神經模糊系統 (Neuro-fuzzy systems, NFSs)。類神經模糊系統是一種模糊系統，它可以說是類神經網絡理論 [29]與模糊理論 [43]的結合，有文獻將 NFS 分類成三種型態 [37]分別是合作型類神經模糊系統(Cooperative NFS)、併發型類神經模糊系統(Concurrent NFS)和混合型類神經模糊系統(Hybrid



NFS)。

**合作型類神經模糊系統:** 是先用類神經網路計算規則等所需參數，在將參數帶入模糊系統中運算。

**併發型類神經模糊系統:** 是數據經過類神經網路運算後，視為模糊系統的輸入再次運算獲得最後結果。

**混合型類神經模糊系統:** 是將模糊系統中的規則等元素融入類神經網路中，並透過學習演算法修正系統中的參數，是目前三者中最被廣泛研究的系統。

由於類神經模糊系統的特性使他對時間序列的預測有著不錯的效果。所以現在有關時間預測這方面的研究，大多採用類神經模糊系統為模型架構。而本研究使用混合式類神經網路，為了使模糊系統所涵蓋的資訊量更加豐富，本研究使用複數型態的模糊集取代傳統的模糊集，形成複數類神經模糊系統(Complex neuro-fuzzy system, CNFS)，使得模型可以同時預測多個目標，細節將會在第三章第一小節中探討。

### 第三章 系統設計與架構

本章節將逐一說明運用於研究中之方法設計與模型架構。本研究使用機器學習決定模型結構的大小，模型實作使用複數類神經模糊系統，分別利用不同的演算法(PSO、ABC)優化模型前鑑部參數，遞迴式最小平方演算法最佳化後鑑部參數。在資料進入模型之前，透過多目標特徵挑選 [28]，挑選出對所有目標較為有效之特徵資料集合，減少龐大資料對模型的負擔。最後將結果配合投資策略做不同演算法的比較。

#### 3.1 複數模糊集

傳統的模糊集合概念 [43]，元素對集合的歸屬程度為一對一的關係。複數歸屬度型態的模糊集合，可以擁有更豐富的歸屬程度，透過此概念可以計算出一筆資料於集合中複數型態的歸屬程度，以便之後模型可以一次預測多個目標。歸屬程度計算流程如下，假設有一複數模糊集合  $S$ ，可以表示如下。

$$S = \{(h, \mu_S(x)) | x \in U\} \quad (2)$$

其中， $\mu_S(x)$  為元素  $x$  的歸屬程度，表示如下。

$$\begin{aligned} \mu_S(x) &= r_S(x) \cdot e^{j\omega_S(x)} \\ &= r_S(x) \cdot \cos(\omega_S(x)) + j \cdot r_S(x) \cdot \sin(\omega_S(x)) \end{aligned} \quad (3)$$

其中， $x$  為字集合  $U$  的數值變數； $r_S(x)$  為振幅函數，是一實數數值介於 0 至 1 間； $\omega_S(x)$  為相位函數，是一實數數值； $j$  為  $\sqrt{-1}$ 。

本實驗採用高斯複數模糊集，此概念由 Li et al. 提出 [24]，為複數模糊集合與高斯函數的結合，使其可以進入模型並分析資料。高斯複數模糊集(Complex Gaussian membership function, cGMF)可以表示如下：

$$\text{cGMF}(x, c, \sigma, \lambda) = r_s(x, c, \sigma) \cdot \exp(j\omega_s(x, c, \sigma, \lambda)) \quad (4)$$

$$\begin{aligned} r_s(x, c, \sigma) &= \text{Gaussian}(x, c, \sigma) \\ &= \exp\left[-0.5 \left(\frac{x - c}{\sigma}\right)^2\right] \end{aligned} \quad (5)$$

$$\begin{aligned}
\omega_s(x, c, \sigma, \lambda) &= -\exp\left[-0.5\left(\frac{x-c}{\sigma}\right)^2\right] \cdot \left[\left(\frac{x-c}{\sigma^2}\right)\right] \cdot \lambda \\
&= -r_s \cdot \left[\left(\frac{x-c}{\sigma^2}\right)\right] \cdot \lambda
\end{aligned} \tag{6}$$

其中， $x, c, \sigma$  分別為輸入資料、中心值以及模糊集合的延展度， $\lambda$  值為此函數的相位頻率參數，此函數會進入參數學習過程，以增加模型整體的彈性。 $\omega_s$  則採用高斯函數的一次微分，目的在於可重複使用高斯原有的參數，降低運算時參數的複雜度。透過複數高斯型態的模糊集，可得出一複數歸屬程度  $\mu_s(x)$ 。我們可以透過拆解  $\mu_s(x)$ ，得出一組歸屬程度向量  $\vec{u} = [u_{x_1} \ u_{x_2} \ u_{x_3}]^T$ ，成分表示如下。

$$u_{x_1} = \mu_s(x) = r_s(x) \cdot \cos(\omega_s(x)) + j \cdot r_s(x) \cdot \sin(\omega_s(x)) \tag{7}$$

$$u_{x_2} = \text{real}(\mu_s(x)) \tag{8}$$

$$u_{x_3} = \text{imag}(\mu_s(x)) \tag{9}$$

其中， $\text{real}(\cdot)$  為擷取該值的實數部位置數值； $\text{imag}(\cdot)$  為擷取該值的實數部位置數值； $r_s(x)$  為公式(5)所提及的高斯函數； $\omega_s(x)$  為公式(6)所提及的高斯函數一次微分。透過上述的拆解可以在不增加參數的情況下，得到有別於傳統模糊集的歸屬程度，此方法提供豐富的資訊量，方便日後做更多元的應用。

## 3.2 結構學習

結構學習是為了透過訓練資料，建造出更適當的模型架構，此外結構學習中的結果，也會成為之後參數學習的一部分。在本研究採用高斯型態的模糊集合，需要中心以及標準差兩個參數，因此會透過減數分群(Subtractive cluster, SC)演算法 [7] 分群不同輸入維度的訓練資料。並將分群後的群中心配合每個維度的標準差形成模糊集，各個維度的模糊集個數總和，即為第 1 層神經元的數量，基於各個輸入維度的模糊集，共可以組成  $B$  個區塊。

$$B = \prod_{j=1}^M \psi_j, \tag{10}$$

其中， $\psi_j$  為第  $j$  個輸入維度的分群個數，亦即模糊集個數。以第  $l$  個區塊的組成為

例，如下。

$Area^{(l)}$  :

$$x_1 = A_1^{(l)}(h_1) \text{ and } x_2 = A_2^{(l)}(h_2) \dots \text{and } x_M = A_M^{(l)}(h_M), \quad (11)$$

$l = 1, 2, \dots, B$ ，其中， $x_j$  為第  $j$  個輸入的語意變數； $h_j$  為第  $j$  個輸入變數， $j = 1, 2, \dots, M$ ； $A_j^{(l)}(\cdot)$  為第  $l$  個區塊中第  $j$  個輸入語意變數的模糊集合，使用高斯函數建構，如公式(5)。

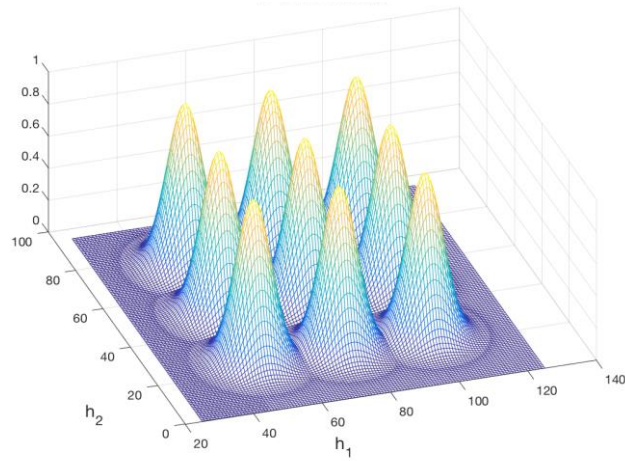


圖 1 模糊集合輸入空間(2 維)

兩個輸入維度，各分出 3 群則會形成共 9 塊區域，其中 z 軸為該區的資料密度。

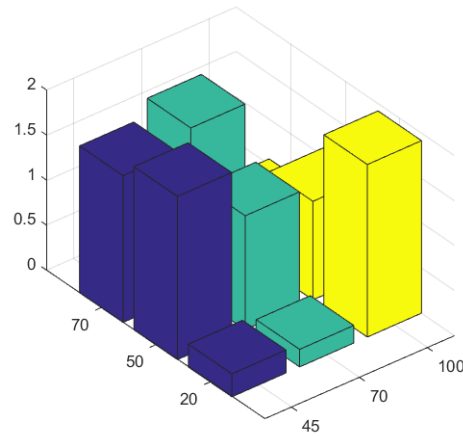


圖 2 資料密度總和

可看出部分區塊資料密度總和較高，代表用於建造模型較有利。

為了降低模型的運算負擔，以提升效率，我們將會篩選出若干個較重要的區塊成為前鑑部神經元，也就是第 2 層中的神經元。若以兩個輸入維度為例子可得

到如圖 1 的區塊，而後透過資料密度的概念，將資料灑入區塊中，計算資料密度量，累加每個區域的資料密度量後可得到如圖 2，從中可挑選資料密度較高的區塊，作為第 2 層神經元。詳細步驟如下：

Step 1. 從各個輸入維度與模糊集可以得到每個區塊的資料密度量，以第  $l$  個區塊為例，資料密度量公式如下。

$$\beta_i^{(l)} = \prod_{j=1}^M A_j^{(l)}(h_j(i)), \quad (12)$$

其中， $h_j(i)$  為第  $j$  個輸入維度的第  $i$  筆資料； $A_j^{(l)}(\cdot)$  為第  $k$  個區塊中第  $j$  個輸入維度的模糊集。

Step 2. 將第  $l$  個區塊的資料密度量累加標記為  $\beta_{\text{sum}}^{(l)}$ ，公式如下。

$$\beta_{\text{sum}}^{(l)} = \sum_{i=1}^n \beta_i^{(l)}, \quad (13)$$

其中， $n$  為資料總筆數。計算  $\{\beta_{\text{sum}}^{(l)}, l = 1, 2, \dots, B\}$  平均值標記為  $\bar{\beta}_{\text{sum}}$ ，標準差標記為  $\beta_{\text{sum}}^{\text{std}}$ 。

Step 3. 查看每個區塊，若  $\beta_{\text{sum}}^{(l)} > \bar{\beta}_{\text{sum}} \beta_{\text{sum}}^{\text{std}}$ ，則將  $b_{\text{tmp}}$  累加。設定上下界，標記為  $b_U$  和  $b_L$ ，透過上下界找出  $b_{\text{FP}}$ ， $b_{\text{FP}}$  表示最後選取的區塊數目。若  $b_{\text{tmp}}$  介於上下界間，則將  $b_{\text{FP}}$  設定成  $b_{\text{tmp}}$ ；若  $b_{\text{tmp}}$  小於下界，則將  $b_{\text{FP}}$  設定成  $b_L$ ；若  $b_{\text{tmp}}$  大於上界則將  $b_{\text{FP}}$  設定成  $b_U$ 。

Step 4. 將  $\{\beta_{\text{sum}}^{(l)}, l = 1, 2, \dots, B\}$  排序，並保留前  $b_{\text{FP}}$  個區塊，當作之後模型第 2 層的神經元。

位於第 4 層の後鑑部神經元個數，在本研究中與前鑑部神經元數目相同。後鑑部神經元為 T-S 神經元，由 T-S function 構成，T-S function 公式如下。

$$y^{(k)} = a_0^{(k)} + \sum_{i=1}^M a_i^{(k)} h_i, \quad (14)$$

其中， $\{a_0^{(k)}, a_i^{(k)}, i = 1, 2, \dots, M\}$  是第  $k$  個 T-S 神經元的參數， $h_i$  是第  $i$  個輸入。

在結構學習後，我們可以透過分群確認第 1 層的神經元個數，藉由區塊挑選

得到 $K$ 個第2層神經元以及第4層神經元個數，藉此創建模型，模型詳細說明將在下一個小節探討。

### 3.3 複數模糊類神經模型

本研究利用 Takagi-Sugeno (T-S) fuzzy 建立模糊系統。T-S 模糊模型最早是由 Takagi 與 Sugeno 於 1985 年提出 [36]，以一個複合式非線性系統並藉由一系列的 If-Then 模糊規則組合而成。本研究是透過複數高斯模糊集非線性的系統，結合線性的 T-S function，形成非線性的 If-Then 模糊規則網路架構。If-Then 模糊規則類似人類的經驗法則，因此更容易被人類所理解。本章節將探討模型各類神經層的輸入、計算方法與輸出。

本研究的模型為一個複合式非線性系統。訓練資料集合標記為  $TD = \{(\vec{h}_i, \vec{t}_i), i = 1, 2, \dots, n\}$ ， $n$  為資料總筆數， $\vec{h}_i$  是  $M$ -by-1 的輸入向量， $M$  為輸入維度數量； $\vec{t}_i$  為  $N$ -by-1 的目標向量， $N$  為複數型態目標的數量。透過模型可以得到輸出  $\{\hat{y}_i, i = 1, 2, \dots, n\}$ 。

模型是透過  $K$  條 T-S 模糊規則組成，每條規則是由前鑑部(If part)與後鑑部(Then part)所結合，其規則形式如下：

$Rule^{(k)}$ :

$$\text{IF } x_1 \text{ is } A_1^{(k)}(h_1) \text{ and } x_2 \text{ is } A_2^{(k)}(h_2) \dots \text{and } x_M \text{ is } A_M^{(k)}(h_M) \quad (15)$$

$$\text{THEN } y^{(k)} = a_0^{(k)} + \sum_{i=1}^M a_i^{(k)} h_i,$$

規則數  $k = 1, 2, \dots, K$ ； $\{h_1, h_2, \dots, h_M\}$  為模糊系統的輸入變數； $\{A_1^{(k)}, A_2^{(k)}, \dots, A_M^{(k)}\}$  為第  $k$  條規則的複數模糊集； $\{x_1, x_2, \dots, x_M\}$  則是輸入的語意變數。複數模糊集  $\{A_i^{(k)}, i = 1, 2, \dots, M\}$  的參數即是前鑑部(If part)參數， $\{a_0^{(k)}, a_1^{(k)}, \dots, a_M^{(k)}\}$  為後鑑部(Then part)參數。此複數類神經模糊模型可轉為一個六層架構的類神經網路模

型，如圖 3，以下將對各層進行說明。

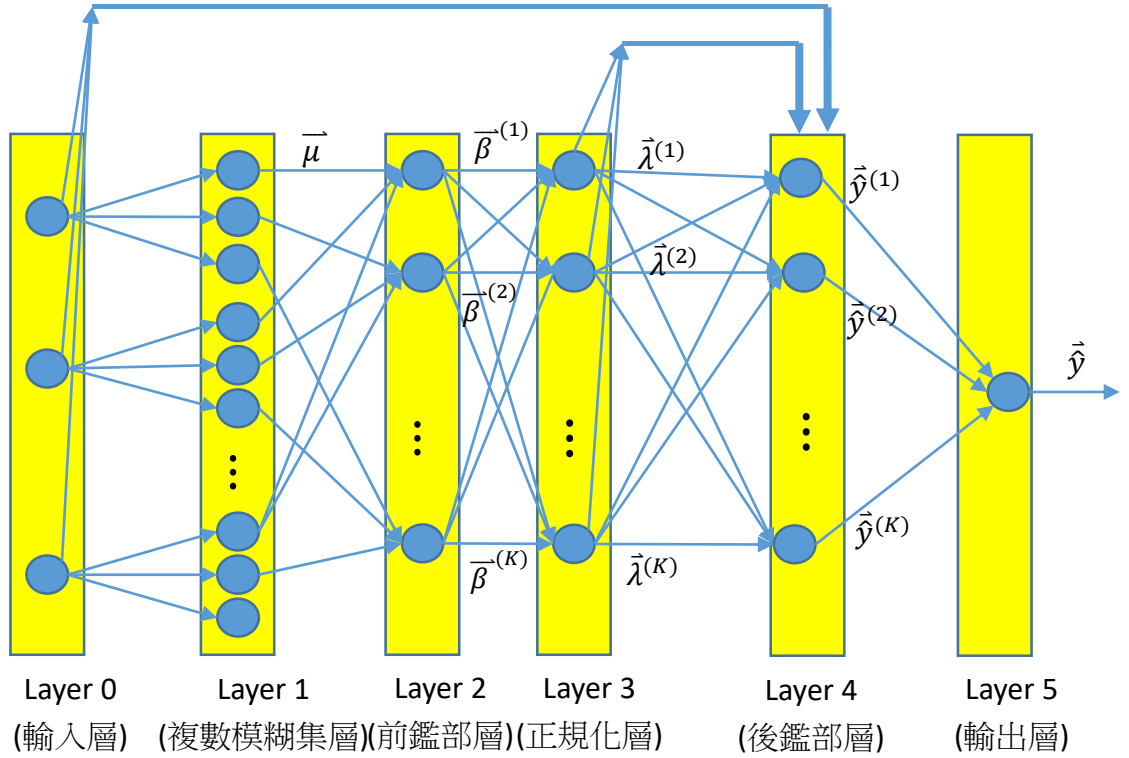


圖 3 複數類神經模糊系統模型

**Layer 0:** 此層為輸入層，是原始資料透過多目標特徵選取後，將最後挑出的特徵當作訓練資料，我們將時間序列第 $i$ 個點的輸入向量標記如下。

$$y^{(k)} = a_0^{(k)} + \sum_{i=1}^M a_i^{(k)} h_i, \quad (16)$$

**Layer 1:** 此層為複數模糊集神經層，透過前面結構學習的分群，可在不同維度上建構數個模糊集，每個不同維度的輸入都可經由模糊集得到歸屬程度。透過複數模糊集合可得到多組複數型態的歸屬程度，不同的歸屬程度可以給不同的模型輸出做應用，以達到多目標預測的效果，透過複數模糊集的公式(2)-(9)，可得到歸屬度向量，如下。

$$\vec{u} = [\mu_1 \quad \mu_2 \quad \mu_3]^T, \quad (17)$$

**Layer 2:** 此層為前鑑部神經層，經過前面的結構學習，我們可以篩選出 $K$ 個對模型較有用的區塊並建構 $\pi$ 神經元，由於 $\pi$ 神經元的輸入為上一層的歸屬程度，且輸出為每個輸入維度的歸屬程度相乘結果，故稱之為 $\pi$ 神經元，每個 $\pi$ 神經元的輸

出為該神經元的啟動強度，由於本研究採用複數模糊集合，因此每個 $\pi$ 神經元的輸入會是向量型態，輸出亦然如此。

$$\vec{\beta}^{(k)} = [\beta_1^{(k)} \quad \beta_2^{(k)} \quad \beta_3^{(k)}]^T, \quad (18)$$

$$\beta_i^{(k)} = \prod_{j=1}^M \mu_{j,i}^{(k)}, \quad (19)$$

$k = 1, 2, \dots, K$ ， $\beta_i^{(k)}$ 為 $\mu_{j,i}^{(k)}$ 的乘積， $\mu_{j,i}^{(k)}$ 為第 $k$ 個 $\pi$ 神經元中第 $j$ 個維度歸屬度向量

$\{\vec{\mu}_j^{(k)} = [\mu_{j,1}^{(k)} \quad \mu_{j,2}^{(k)} \quad \mu_{j,3}^{(k)} \quad \mu_{j,4}^{(k)}]\}$ 的第 $i$ 項歸屬程度， $i = 1, 2, 3$ 。

**Layer 3:** 此層為正規化層，會將 $K$ 個輸入向量中的各個元素正規化，之後將結果以向量型態輸出，輸出如下。

$$\vec{\lambda}^{(k)} = [\lambda_1^{(k)} \quad \lambda_2^{(k)} \quad \lambda_3^{(k)}]^T, \quad (20)$$

$$\lambda_i^{(k)} = \frac{\beta_i^{(k)}}{\sum_{k=1}^K \beta_i^{(k)}} \quad (21)$$

其中， $\lambda_i^{(k)}$ 為第 $k$ 個 $\pi$ 神經元中第 $i$ 個元素正規化後的值， $i = 1, 2, 3$ 。本研究中使用的是複數模糊集，因此輸入是複數型態，故輸出也是複數型態。

**Layer 4:** 此層為後鑑部層，經過此層的運算可以得到 $K$ 個模型輸出，公式如下。

$$\vec{\hat{y}}^{(k)} = \vec{\lambda}^{(k)}(a_0^{(k)} + \sum_{i=1}^M a_i^{(k)} h_i), \quad (22)$$

$k = 1, 2, \dots, K$ ， $\vec{\hat{y}}^{(k)}$ 為第 $k$ 個 T-S 神經元輸出； $\{a_0^{(q)}, a_i^{(q)}, i = 1, 2, \dots, M\}$ 是第 $k$ 個 T-S 神經元的參數。

**Layer 5:** 此層為輸出層，將上一層得到的 $K$ 個神經元輸出加總，即為我們的模型輸出。

$$\vec{\hat{y}} = \sum_{q=1}^Q \vec{\hat{y}}^{(q)} \quad (23)$$



### 3.4 參數學習演算法

根據分治法(Divide-and-conquer)的概念，我們將使用不同的機器學習演算法，對各層的參數優化，以便更容易找到最佳解。對於第 1 層複數模糊集的參數優化，我們使用兩種不同的演算法，其中包含粒子群演算法(Particle swarm optimization, PSO) [19]以及人工蜂群演算法(Artificial bee colony optimization, ABCO) [17]。第 2 層與第 3 層沒有參數需要優化。在第 4 層的部分，採用遞迴最小平方演算法(Recursive least squares estimation, RLSE) [16]最佳化後鑑部參數，希望透過不同演算法的結合，降低所需搜尋的參數維度，為模型帶來更好的效能表現。以下將會詳細敘述三種演算法的運作原理以及公式等細節。

#### 3.4.1 粒子群演算法

粒子群演算法是由 J. Kennedy et al. [19]於 1995 年開發的一種演化計算技術，來源於對一個簡化社會模型的模擬。其原理類似鳥群在尋找食物，除了自身提供的資訊，慣性以及自身最佳位置，也運用到群體智慧中全群最佳位置，用以調節速度，如圖 4，此演算法特性為收斂快速，公式如下。

$$\vec{V}_i(t+1) = \omega \vec{V}_i(t) + c_1 \xi_1 (\overrightarrow{Pbest}_i(t) - \vec{P}_i(t)) + c_2 \xi_2 (\overrightarrow{Gbest}(t) - \vec{P}_i(t)), \quad (24)$$

$$\vec{P}_i(t+1) = \vec{P}_i(t) + \vec{V}_i(t+1), \quad (25)$$

其中， $\vec{P}_i(t)$ 為第 $t$ 回合時第 $i$ 個粒子的位置， $\vec{V}_i(t)$ 為第 $t$ 回合時第 $i$ 個粒子的速度， $\overrightarrow{Pbest}_i(t)$ 為第 $t$ 回合時第 $i$ 個粒子的最好位置， $\overrightarrow{Gbest}(t)$ 為第 $t$ 回合時全部粒子中最好的位置， $\omega$ 、 $C_1$ 、 $C_2$ 為 PSO 的參數， $\xi_1$ 、 $\xi_2$ 為介於 0 到 1 的隨機數。在本實驗中，粒子的位置代表前鑑部的參數，其中包含了每個維度的分群中心、標準差以及相位頻率參數。

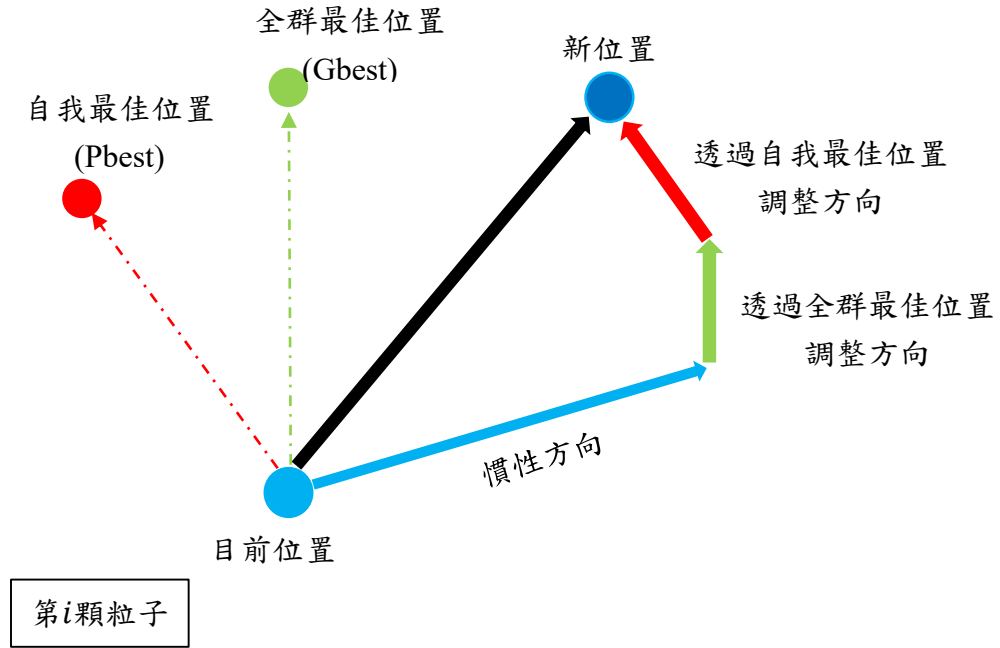


圖 4 PSO 粒子更新位置示意圖

### 3.4.2 人工蜂群演算法

人工蜂群演算法為 Karaboga 學者所提出 [17]，原理類似蜜蜂尋找食物來源的概念，其特性包含群體智慧與隨機性，如蜜蜂傳達食物位置時會彼此透過搖擺舞溝通，而搖擺舞所指示食物位置會有偏差帶有隨機性。此演算法中，總共有三種蜜蜂，包括工蜂(Employed bee)、觀察蜂(Onlooker bee)以及偵查蜂(Scout bee)，工蜂負責對每個食物源的附近進行開發，然後以跳舞傳達食物源的收益訊息，每隻工蜂代表著一食物源，代表著大維度的搜尋；觀察蜂觀察各個食物收益度後，以輪盤法選擇一個食物源，到食物源的鄰近周遭進行開發的行為，代表著小維度的搜尋；偵查蜂作用為當食物源經過一定的開發次數後，食物源的收益度沒有改善時，會派出偵查蜂取代食物源，意味著隨機的探索一個全新的食物源，可以避免演算法陷入區域最佳解中。其步驟與真實的蜜蜂找尋食物不盡相同，演算法步驟如下：

Step 1. 隨機形成工蜂位置，並更新工蜂位置，公式如下。

$$v_i^j = x_i^j + rand[-1,1](x_i^j - x_k^j), \quad (26)$$

其中， $v_i^j$  為第  $i$  隻工蜂在移動後的第  $j$  個維度； $x_i^j$  為第  $i$  隻工蜂移動前的第  $j$  個維度； $x_k^j$  為其他隨機蜜蜂的第  $j$  個維度的值。

Step 2. 使用輪盤法(Roulette method)選一個工蜂的位置，輪盤機率公式如下。

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \quad (27)$$

其中， $p_i$  為第  $i$  隻工蜂被選中的機率； $fit_i$  為適應值，本實驗將成本函數的倒數視為適應值； $SN$  為工蜂的總數目

Step 3. 所有的觀察蜂在剛剛被選中的食物源位置附近搜尋，公式如下。

$$v_i^j = x_i^j + rand[-1,1](x_i^j - x_{selected}^j) \quad (28)$$

其中， $v_i^j$  為第  $i$  隻工蜂在移動後的第  $j$  個維度； $x_i^j$  為第  $i$  隻工蜂移動前的第  $j$  個維度； $x_{selected}^j$  為被選中蜜蜂的第  $j$  個維度的值。

Step 4. 判斷每隻工蜂是否已經達到限制回合都未更新，若為真則派出偵查蜂取代，偵查蜂位置產生公式如下。

$$x_i^j = x_{min}^j + rand[0,1](x_{max}^j - x_{min}^j) \quad (29)$$

其中， $x_i^j$  為第  $i$  隻工蜂的第  $j$  個維度； $x_{max}^j$  為所有工蜂第  $j$  個維度的最大值； $x_{min}^j$  為所有工蜂第  $j$  個維度的最小值。

Step 5. 重複 step 2~ step 4，直到反覆運算結束。

### 3.4.3 遞迴最小平方演算法

本研究使用遞迴式最小平方演算法(Recursive least square estimation, RLSE) [16]更新 T-S 神經元參數，RLSE 方法在更新參數時是利用每筆資料，不斷的更新比起一次接收所有資料的 LSE 方法更加有效，一般來說 LSE 問題可以被視為一個線性的問題，如下。

$$y = f_1(u)\theta_1 + f_2(u)\theta_2 + \cdots + f_m(u)\theta_m + \varepsilon, \quad (30)$$

其中， $y$  是目標; $u$  是模型的輸出; $\{f_i(\cdot), i = 1, 2, \dots, m\}$  是  $u$  已知的方程式; $\{\theta_i, i = 1, 2, \dots, m\}$  是我們估計的未知參數， $\varepsilon$  則是整個模型的誤差 LSE 的問題也可以被寫成矩陣的方式表達，如下。

$$\mathbf{Y} = \mathbf{C}\boldsymbol{\theta} + \varepsilon, \quad (31)$$

其中，

$$\mathbf{C} = \begin{bmatrix} f_1(u_1) & f_2(u_1) & \cdots & f_m(u_1) \\ f_1(u_2) & f_2(u_2) & \cdots & f_m(u_2) \\ \vdots & \vdots & \cdots & \vdots \\ f_1(u_N) & f_2(u_N) & \cdots & f_m(u_N) \end{bmatrix}, \quad (32)$$

$$\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_m]^T, \quad (33)$$

$$\mathbf{Y} = [y_1, y_2, \dots, y_N]^T, \quad (34)$$

$$\varepsilon = [\varepsilon_1 \quad \varepsilon_2 \quad \cdots \quad \varepsilon_N]^T, \quad (35)$$

$\mathbf{C}$  是輸入的矩陣， $\boldsymbol{\theta}$  是我們估計的未知參數矩陣， $\mathbf{Y}$  是目標矩陣， $\varepsilon$  是誤差的向量。

要最佳化  $\boldsymbol{\theta}$ ，可透過 RLSE 的等式[17]運算。

$$P_{r+1} = P_r - \frac{P_r b_{r+1} (b_{r+1})^T P_r}{1 + (b_{r+1})^T P_r b_{r+1}}, \quad (36)$$

$$\boldsymbol{\theta}_{r+1} = \boldsymbol{\theta}_r + P_{r+1} b_{r+1} (y_{r+1} - (b_{r+1})^T \boldsymbol{\theta}_r), \quad (37)$$

其中， $r$  是遞迴次數， $\{r = 0, 1, \dots, (n-1)\}$ ， $n$  為資料總筆數， $[(b_r)^T, y_r]$  是  $[\mathbf{C}, \mathbf{Y}]$  的第  $r$  行，再開始 RLSE 演算法時，會設定  $\boldsymbol{\theta}_0$  為 0， $\mathbf{P}_0$  則設定為  $\alpha \mathbf{I}$ ， $\alpha$  為一極大整數， $\mathbf{I}$  為單位矩陣。

混合型演算法中各部分參數優化會交由不同演算法負責，本研究使用到 PSO-RLSE 以及 ABCO-RLSE 兩種混合型演算法，PSO 和 ABCO 負責前鑑部的參數學習，RLSE 則負責線性的 T-S function 參數學習。混合演算法與模型計算的流程如下：

Step 1. 準備訓練資料及測試資料。

Step 2. 以前鑑部演算法(PSO 或 ABCO)粒子位置作為模糊集參數，將訓練資料帶入模型，並計算每個  $\pi$  神經元啟動強度。

Step 3. 用 RLSE 更新 T-S 神經元的參數，RLSE 算式中的  $b_{r+1}$  和  $\boldsymbol{\theta}_r$  向量如

下。

$$\mathbf{b}_{r+1} = [{}^1bb(r+1) \quad {}^2bb(r+1) \quad \dots \quad {}^Kbb(r+1)], \quad (38)$$

$${}^kbb(r+1) = [\vec{\lambda}^{(k)} \quad \vec{\lambda}^{(k)}h_1(r+1) \quad \dots \quad \vec{\lambda}^{(k)}h_M(r+1)], \quad (39)$$

$$\theta_r = [\tau^{(1)}(r) \quad \tau^{(2)}(r) \quad \dots \quad \tau^{(K)}(r)]^T, \quad (40)$$

$$\tau^{(k)}(r) = [a_0^{(k)}(r) \quad a_1^{(k)}(r) \quad \dots \quad a_2^{(k)}(r)]^T, \quad (41)$$

其中  $r = 1, 2, \dots, (n-1), k = 1, 2, \dots, K$ 。因為多目標預測中，正規化後的啟動強度為一向量，使  $\mathbf{b}_{r+1}$  為一矩陣，因此在原本的公式(36)中，利用單位矩陣取代原本的常數項 1，改良後公式如下。

$$\mathbf{P}_{r+1} = \mathbf{P}_r - \frac{\mathbf{P}_r \mathbf{b}_{r+1}}{\mathbf{I} + (\mathbf{b}_{r+1})^T \mathbf{P}_r \mathbf{b}_{r+1}} (\mathbf{b}_{r+1})^T \mathbf{P}_r, \quad (42)$$

Step 4. 更新完所有參數後，計算出模型的輸出。

Step 5. 計算成本，更新前鑑部演算法粒子的位置及相關數據。

Step 6. 重複 Step 2~Step 5，直到迭代結束。

### 3.5 投資策略

為了評估模型是否對投資有實質上的幫助，使用成本函數是不夠的，因為從中無法看出是否有利潤，只能瞭解模型的配市率，而配市率高並不代表投資效益高，因此本實驗將預測出來的收盤價配合投資策略 [38]，進一步決定要買進或是賣出，買進與賣出公式如下。

$$\text{買進: if } \frac{|forecast(t) - actual(t)|}{actual(t)} \leq \varphi \text{ and } forecast(t+1) - actual(t) > 0, \quad (43)$$

$$\text{賣出: if } \frac{|forecast(t) - actual(t)|}{actual(t)} \leq \varphi \text{ and } forecast(t+1) - actual(t) < 0, \quad (44)$$

其中， $\varphi$  為門檻參數，同時也代表股票的漲跌； $forecast(t)$  為模型的輸出，意即預測  $t$  日的收盤價格； $actual(t)$  為  $t$  日實際的收盤價格。若預測明天的收盤價高於今天實際收盤價，代表必須買進；若預測明天的收盤價低於今天實際收盤價，代表模型預測明天會跌，所以要儘快賣出。

計算利潤的方式，則透過今天實際的收盤價與隔天實際的收盤價做運算，公式如下。

$$\begin{aligned} Profit = & \sum_{t_b=1}^p (actual(t) - actual(t+1)) \\ & + \sum_{t_s=1}^q (actual(t+1) - actual(t)) \end{aligned} \quad (45)$$

其中， $Profit$ 為利潤， $p$ 為策略為買的總天數； $q$ 為策略為賣的總天數； $actual(t)$ 代表第 $t$ 天的收盤價。

透過上述投資策略以及利潤公式，我們可以計算出整個模型的所帶來的利潤值，並大致模擬出此模型運用到真實世界的效果。本研究將會在每個實驗中秀出利潤值與其他參數。

## 第四章 實驗

本研究總共有 3 個實驗，實驗一為單目標的預測，使用的目標為 2001 年台灣股票加權指數(The Taiwan stock exchange capitalization weighted stock index, TAIEX)；實驗二為雙目標的預測，檢驗模型透過複數型態的歸屬程度可行性，使用第一組複數輸出預測兩個目標，實數部分負責第一個目標，虛數部分負責第二個目標，同時預測 TAIEX 以及恆生指數(Hang seng index, HSI)；實驗三為四個目標的多目標預測，使用到兩組複數型態的輸出，比實驗二複雜度更高，其中包含了 2001 年的台股指數、道瓊工業指數(Dow Jones industrial average index, DJI)、納斯達克(National association of securities dealers automated quotation, NASDAQ)以及標準普爾 500(Standard and Poor's 500, S&P 500)。上述目標皆為非常著名的股票指標，像是 TAIEX 為台灣上市的股票中經過加權計算出的指標，代表著台灣上市股票的波動；HSI 是以反映香港股市行情的重要指標，指數由五十隻恆指成份股的市值計算出來的，相當於香港交易所所有上市公司十二個月平均市值涵蓋率的 63%；DJI 涵蓋著財務等 9 大產業，為一股價加權指標；NASDAQ 為超過三千檔股票所組合成的市值加權指標，大多以科技產業為例；S&P500 為美國前 500 大公司的市值加權，當中包含 IT 等 11 個產業。這些指標代表著一個國家甚至全球股票的趨勢，故能精準預測可為投資者帶來不少的幫助。

為了與其他文獻比較模型的好壞，我們將透過誤差指標對模型評估以及計算模擬投資後的利潤。成本函數(Cost function)與評估指標皆使用均方根誤差(Root mean square errors, RMSE)，公式如下。

$$\vec{e}_i = \vec{y}_i - \vec{\hat{y}}_i, \quad (46)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (\vec{e}_i)^* (\vec{e}_i)}{n}}, \quad (47)$$

$i = 1, 2, \dots, n$ ， $n$  為資料總筆數； $\vec{e}_i$  為模型第  $i$  筆資料的誤差向量； $\vec{y}_i$  為第  $i$  筆目標向量； $\vec{\hat{y}}_i$  為第  $i$  筆模型輸出向量； $(\cdot)^*$  為埃爾米特共軛(Hermitian transpose)，意即轉置

矩陣後，並對矩陣元素做共軛運算。

在結構學習部分，本研究所有實驗的上限 $b_U$ 皆設定為 15，下限 $b_L$ 皆設定為 4，故模型的規則數會介於 4~15 之間。

在模擬投資方面，所有實驗的 $\varphi$ 會介於 0 至 0.1 之間，因為 $\varphi$ 代表著股票的波動，而台灣股票的漲跌幅為 10%，因此會從此區間中挑選出最佳的 $\varphi$ ， $\varphi$ 從 0 開始每次以 0.001 增加，直至 0.1。

#### 4.1 實驗一：台灣股票加權指數單目標預測

本實驗使用真實世界的時間序列數據來驗證模型的效能，目標為 TAIEX。實驗目標是建立一適當的模型，經過訓練後預測每日股價指數。模型輸出為一複數值，取其實數部分預測目標。使用的資料為 TAIEX 2001 年的每日收盤價，此實驗的原始資料為 278 筆，做一次差分後，從中擷取出 30 個特徵，每個特徵有 247 筆資料，前 181 筆資料作為訓練資料，剩餘作為測試資料。經過多目標特徵挑選後選出特徵 $\{f_3, f_4, f_{15}, f_6\}$ 做為模型輸入資料。結構學習部分，會透過減數分群演算法分群，在進行區塊挑選如第三章第二小節所介紹，本實驗各個輸入維度分別有 $\{3, 3, 3, 3\}$ 個複數模糊集，進行區塊挑選後，從原本的 144 個 $\pi$ 神經元篩選到剩下 15 個 $\pi$ 神經元，藉由資料驅動(Data driven)概念適當地建構模型，每個複數模糊集有 3 個參數，其中包含中心、標準差以及相位頻率參數，總共有 12 個複數模糊集，故前鑑部參數數量為 36，後鑑部型態為 T-S function，因此總共會有 $K * (M + 1)$ 個參數， $K$ 為後鑑部數目， $M$ 為模型輸入維度數目。整體模型在結構學習後的參數，如表 1 所示。機器學習參數設定，如表 2 所示。本次範例的結果除了兩個演算法比較也與其他論文 [38]所提的方法做比較，像是 Chen、Yu、SVR 和 ANFIS。比較結果如表 3 所示。為了測試模型的穩定性，我們總共跑了十次的實驗，效能統計結果如表 4 所示，可以從中發現有訓練過的模型較可能尋找到更佳的模型參數。模擬投資所賺的利潤，以及買賣次數如表 5 所示，有長時間訓練的



模型，擁有較低的標準差代表穩定性較佳，相對來說標準差越高，代表投資風險越高。模擬投資的利潤比較如表 6 所示。目標與模型輸出的結果，如圖 5 所示；模型的機器學習曲線，如圖 6 所示；預測誤差如圖 7 所示。

表 1 實驗一模型設定

參數	值
特徵變數(輸入變數)	$\{f_3, f_4, f_{15}, f_6\}$
模糊集數量	$\{3, 3, 3, 3\}$
模糊集型態	CFS
複數型態目標數目	1
$\Pi$ 神經元數目	15
前鑑部參數數目	36
T-S 神經元數目	15
後鑑部參數數目	75

\* 每一個複數目標的實部與虛部，分別包含兩個實數目標

表 2 機器學習參數設定

<b>PSO</b>	
粒子群大小	50
迭代次數	100
$\{\omega, c_1, c_2\}$	$\{0.8, 2.0, 2.0\}$
$\xi_1, \xi_2$	Random in $[0,1]$
粒子初始化位置	藉由第三章所提 SC 演算法決定
粒子初始化速度	0
<b>ABCO</b>	
工蜂數量	40
觀察蜂數量	10
迭代次數	100
限制回合數	20
<b>RLSE</b>	
$\alpha$	$1 \times 10^8$
$\bar{\theta}_0$	25-by-1 全零向量
$P_0$	$\alpha I$
$I$	25-by-25 單位矩陣

表 3 效能比較(TAIEX, 實驗一)

方法	RMSE
Chen [38]	167
Yu [38]	148
AR(1) [38]	115
SVR [38]	114
ANFIS [38]	120
Wei [38]	110
PSO-RLSE (proposed)	102.01
ABCO-RLSE (proposed)	<b>101.93</b>

表 4 十次重複實驗效能統計(TAIEX, 實驗一)

Trials	PSO-RLSE	ABCO-RLSE	PSO-RLSE*	ABCO-RLSE*
1	102.33	102.94	102.47	103.38
2	102.17	105.22	102.61	102.75
3	104.88	102.69	103.15	108.72
4	102.69	102.75	102.59	102.39
5	102.96	<b>101.93</b>	102.05	103.12
6	<b>102.01</b>	102.97	<b>102.03</b>	107.92
7	103.17	103.15	102.61	<b>102.38</b>
8	117.92	103.01	103.35	129.42
9	102.54	104.63	106.68	102.63
10	103.96	106.87	102.96	103.02

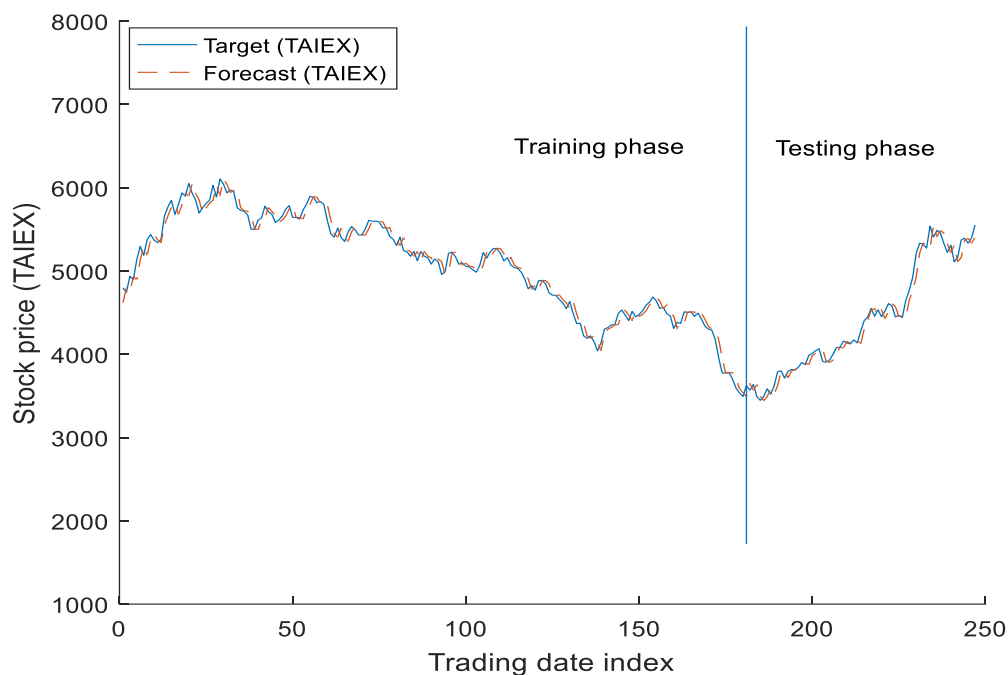
表 5 模擬投資利潤表(實驗一)

Trials	PSO-RLSE		ABCO-RLSE		PSO-RLSE*		ABCO-RLSE*	
	利潤	買/賣	利潤	買/賣	利潤	買/賣	利潤	買/賣
Best $\varphi$	0.005		0.005		0.005		0.005	
1	<b>355.64</b>	15/21	198.86	16/20	355.64	15/21	401.74	14/20
2	312.45	16/18	<b>420.35</b>	16/21	355.64	15/21	272.29	15/20
3	152.16	18/19	355.64	15/21	-117.1	18/18	292.45	16/21
4	108.31	18/21	263.33	15/21	355.64	15/21	355.64	15/21
5	83.29	19/19	236.87	16/19	355.64	15/21	404.49	16/21
6	47.72	15/22	180.20	16/22	<b>422.37</b>	14/21	280.75	17/19
7	139.12	16/19	392.04	15/21	355.64	15/21	355.64	15/21
8	-307.6	17/21	209.07	19/19	-172.7	18/16	327.68	13/19
9	147.63	16/20	-33.91	16/20	355.00	14/17	<b>646.08</b>	19/19
10	154.35	17/18	-124.5	19/20	258.03	15/21	235.92	17/21
平均	119.29		209.79		252.38		357.27	
標準差	178.12		174.67		213.44		115.71	

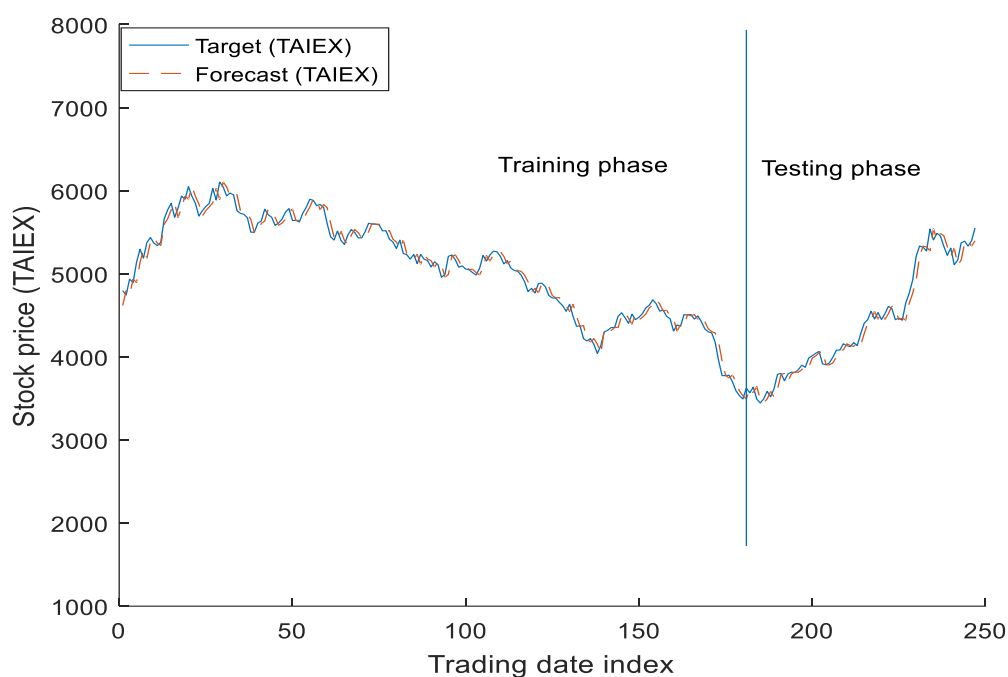
\*為只訓練一個迭代的模型

表 6 模擬投資利潤比較表(實驗一)

	Chen [38]	Yu [38]	AR(1) [38]	SVR [38]	ANFIS [38]	Wei [38]	PSO-RLSE	ABCO-RLSE
Best $\varphi$	0.01	0.01	0.01	0.01	0.01	0.01	0.005	0.005
利潤(TAIEX)	-92	-73	671	202	686	<b>795</b>	355.64	422.37



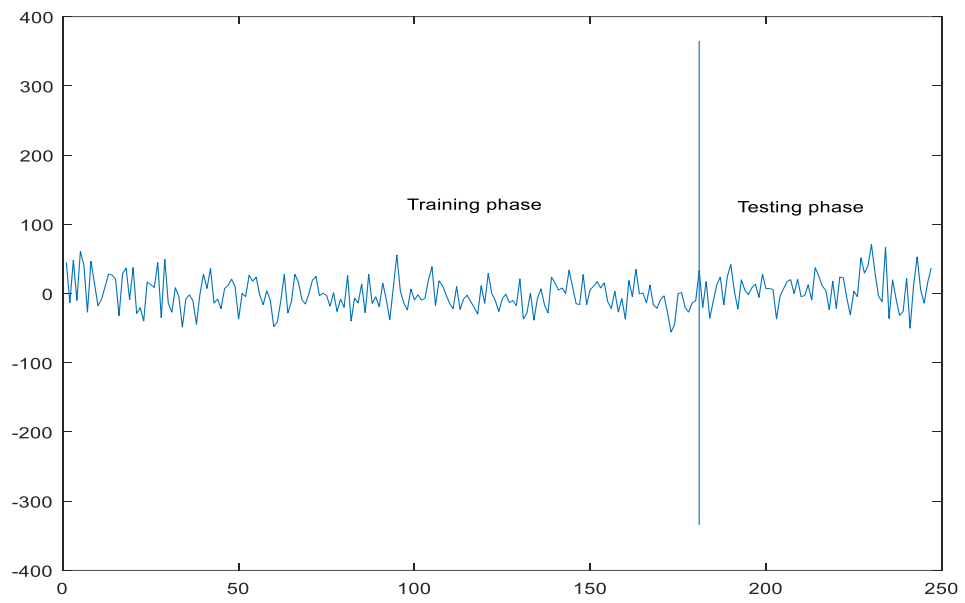
(a) TAIEX 預測(PSO-RLSE)



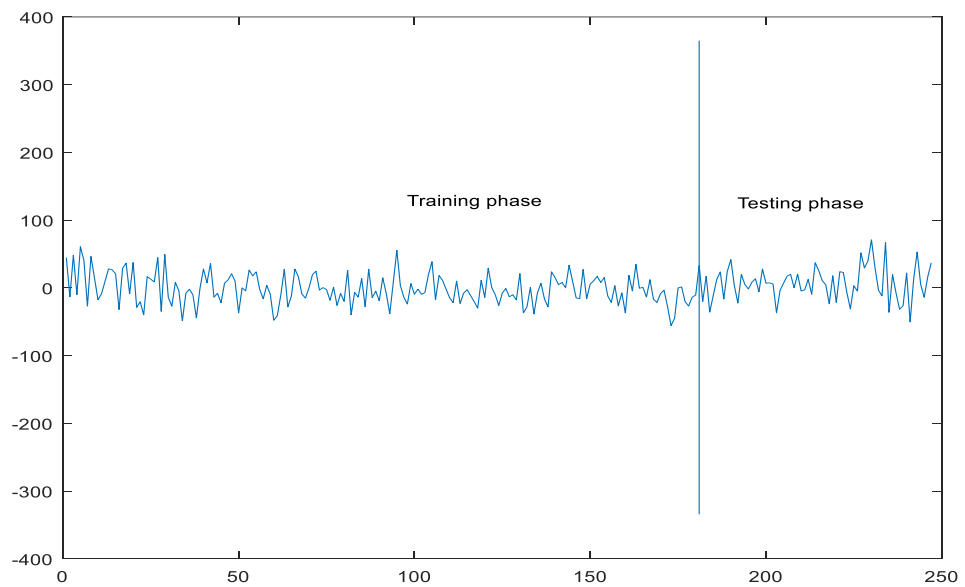
(b) TAIEX 預測(ABCO-RLSE)

圖 5 預測結果

藍色實線為實際目標值，紅色虛線為模型預測值，在測試與訓練階段看起來都有著不錯的成果。



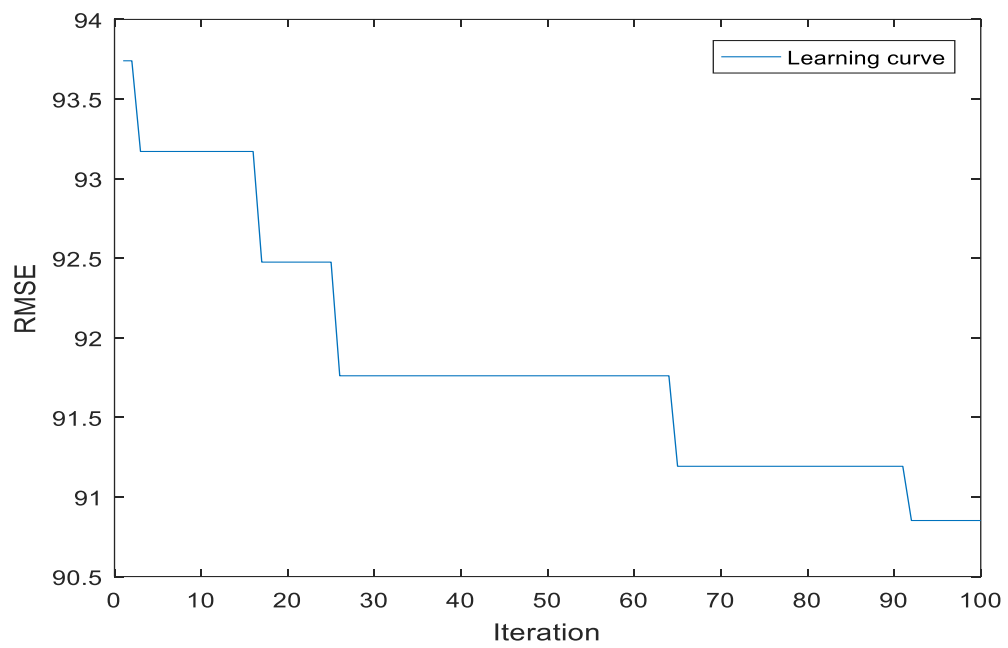
(a) PSO-RLSE



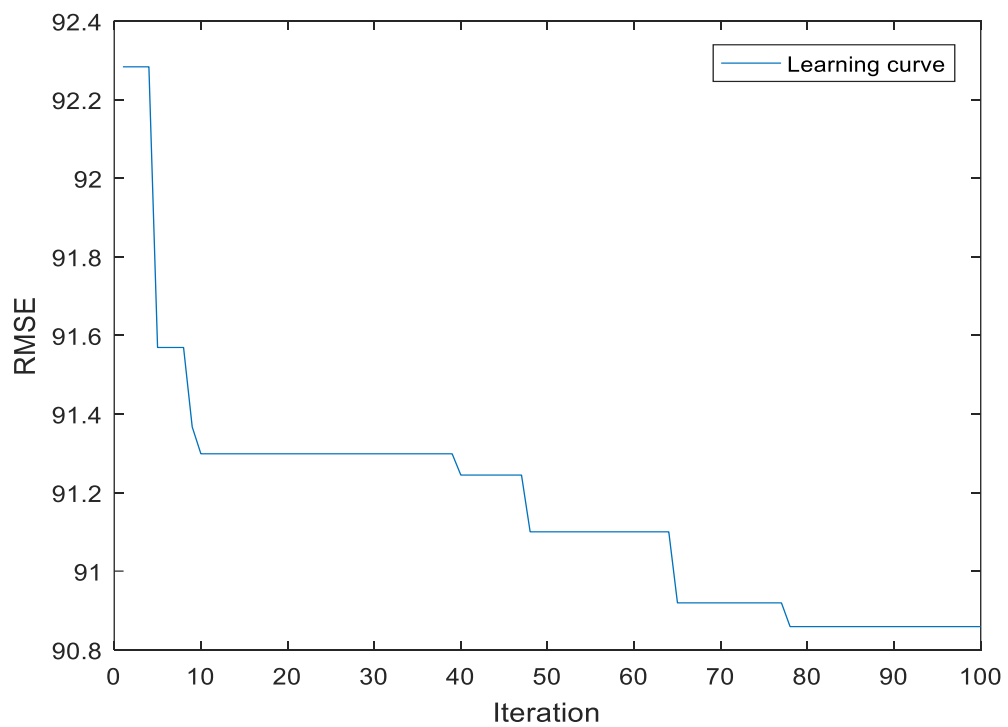
(b) ABCO-RLSE

圖 6 模型預測誤差值

誤差值兩個演算法都以亂數分布呈現，介於-50~50 之間，代表模型預測能力穩定



(a) PSO-RLSE



(b) ABCO-RLSE

圖 7 實驗一模型學習曲線

## 4.2 實驗二：台灣股票加權指數與恆生指數雙目標預測

本實驗一次預測兩個目標，分別為 TAIEX 與 HSI。實驗目標是建立一適當的模型，經過訓練後預測每日股價指數。模型輸出為一複數值，取其實數部分預測 TAIEX，虛數部分預測 HSI。使用的資料為 TAIEX 2000 年的每日收盤價以及 HIS 2000 年的每日收盤價，此實驗的原始資料為 277 筆，經過一次差分後得到 276 筆，從每組資料中取出 30 個特徵值，共 60 個特徵，每個特徵為 246 筆資料，其中前 204 筆為訓練資料，剩餘為訓練資料，第 1 至 30 個特徵為 TAIEX 收盤價，第 31 至 60 為 HIS 收盤價。透過多目標特徵挑選後選出特徵 $\{f_{33}, f_{19}, f_1, f_{31}\}$ 做為模型輸入資料。結構學習部分，會透過減數分群演算法分群，在進行區塊挑選如第三章第二小節所介紹，本實驗各個輸入維度分別有 $\{3, 3, 3, 3\}$ 個複數模糊集，進行區塊挑選後，從原本的 81 個 $\pi$ 神經元篩選到剩下 9 個 $\pi$ 神經元，藉由資料驅動(Data driven)概念適當地建構模型，每個複數模糊集有 3 個參數，其中包含中心、標準差以及相位頻率參數，總共有 12 個複數模糊集，故前鑑部參數數量為 36，後鑑部型態為 T-S function，因此總共會有 $K * (M + 1)$ 個參數， $K$ 為後鑑部數目， $M$ 為模型輸入維度數目。整體模型在結構學習後的參數，如表 7 所示。機器學習參數設定，如表 8 所示。本次範例的結果除了兩個演算法比較也與其他論文[6]所提的方法做比較，像是 Chen、Yu、SR+ANFIS 等。比較結果如表 9 所示。為了測試模型的穩定性，我們總共跑了十次的實驗，效能統計結果如表 10 所示，可以發現訓練過後的模型容易達到更好的效能。模擬投資所賺的利潤，以及買賣次數如表 11 所示，訓練過後模型的標準差較低，代表投資風險較低。模擬投資利潤比較表如表 12 所示。目標與模型輸出的結果，如圖 8 所示；模型的機器學習曲線，如圖 9 所示；預測誤差如圖 10 所示。

表 7 實驗二模型設定

參數	值
特徵變數(輸入變數)	$\{f_{33}, f_{19}, f_1, f_{31}\}$
模糊集數量	$\{2, 3, 3, 3\}$
模糊集型態	CFS
複數型態目標數目	1
$\Pi$ 神經元數目	9
前鑑部參數數目	36
T-S 神經元數目	9
後鑑部參數數目	45

\* 每一個複數目標的實部與虛部，分別包含兩個實數目標

表 8 機器學習參數設定

<b>PSO</b>	
粒子群大小	50
迭代次數	100
$\{\omega, c_1, c_2\}$	$\{0.8, 2.0, 2.0\}$
$\xi_1, \xi_2$	Random in $[0,1]$
粒子初始化位置	藉由第三章所提 SC 演算法決定
粒子初始化速度	0
<b>ABCO</b>	
工蜂數量	40
觀察蜂數量	10
迭代次數	100
限制回合數	20
<b>RLSE</b>	
$\alpha$	$1 \times 10^8$
$\bar{\theta}_0$	25-by-1 全零向量
$P_0$	$\alpha I$
$I$	25-by-25 單位矩陣

表 9 效能比較(實驗二)

方法	TAIEX	HSI
Chen [6]	413.27	280.15
Yu [6]	419.64	297.05
SR+ANFIS [6]	454.63	356.70
SR+SVR [6]	255.87	356.81
Elman [6]	154.21	302.27
Cheng [6]	<b>150.55</b>	251.70
PSO-RLSE	151.06	254.97
ABCO-RLSE	153.45	<b>250.51</b>

表 10 十次重複實驗效能統計(實驗二)

Trials	PSO-RLSE	ABCO-RLSE	PSO-RLSE*	ABCO-RLSE*
1	298.49	296.49	<b>295.88</b>	296.96
2	<b>296.36</b>	304.76	297.66	298.59
3	302.48	296.26	310.14	297.31
4	298.20	296.25	296.95	<b>294.17</b>
5	305.08	296.59	312.79	296.23
6	332.39	<b>293.77</b>	303.49	297.90
7	308.42	314.39	295.99	297.14
8	303.56	295.34	296.53	296.32
9	297.36	297.58	297.65	294.17
10	329.41	307.30	296.38	296.50

表 11 模擬投資利潤表(實驗二)

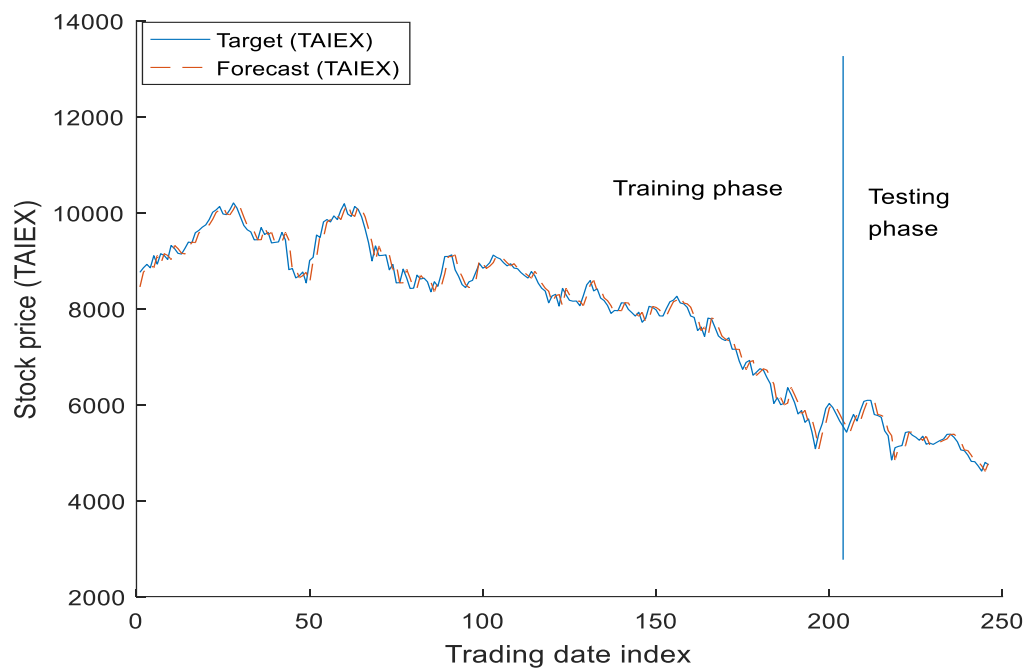
Trials	PSO-RLSE		ABCO-RLSE		PSO-RLSE*		ABCO-RLSE*	
	利潤	買/賣	利潤	買/賣	利潤	買/賣	利潤	買/賣
Best $\varphi$	0.002		0.002		0.002		0.004	
1	<b>1038.1</b>	3/3	782.49	2/3	1038.1	3/3	781.17	6/9
2	939.21	3/5	773.54	4/5	360.9	3/5	975.57	5/11
3	-15.41	3/3	687.92	4/3	-492.2	5/5	1475.0	5/11
4	824.60	3/3	586.72	3/5	1029.5	4/4	<b>1652.5</b>	4/10
5	-289.1	4/3	449.34	4/3	-335.9	5/1	778.89	8/8
6	683.61	2/2	57.12	5/4	350.62	4/4	482.06	6/8
7	584.73	3/2	912.85	3/4	<b>1227.3</b>	3/5	735.97	5/12
8	-344.7	4/3	683.61	2/2	939.23	4/3	1231.5	8/11
9	609.60	4/5	<b>1227.3</b>	3/5	-119.8	5/4	1485.3	6/10
10	176.25	6/2	614.87	2/4	1137.0	3/4	1201.6	4/8
平均	420.67		677.57		513.47		1080	
標準差	503.82		303.15		650.18		387.66	

\*為只訓練一個迭代的模型

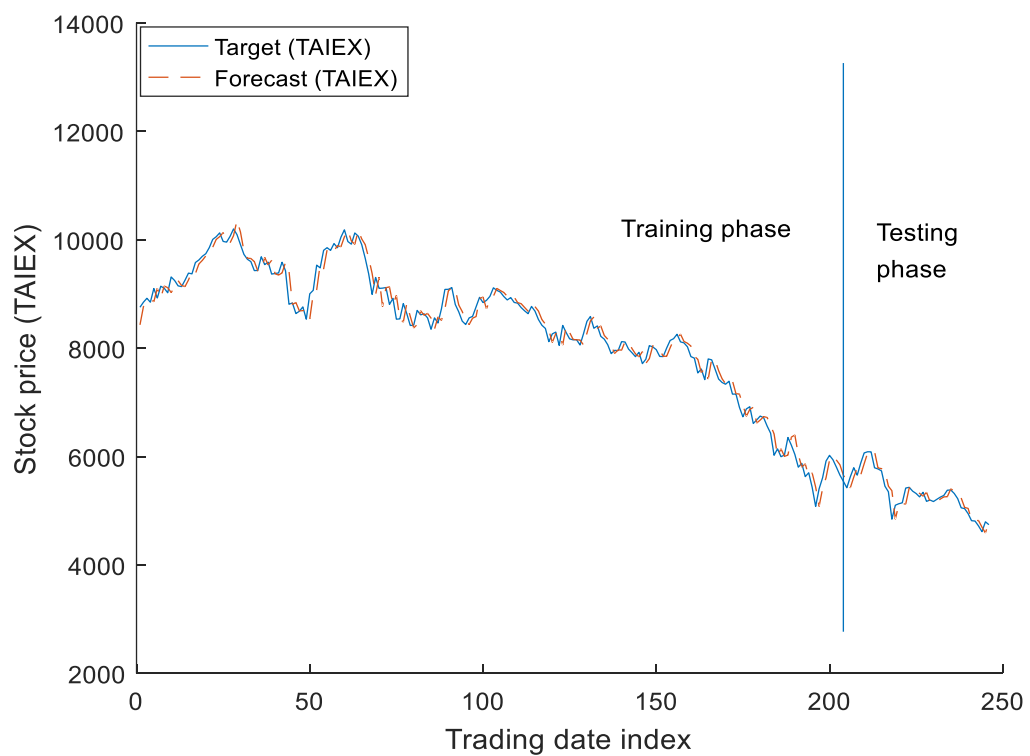
表 12 模擬投資利潤比較表(實驗二)

	Chen [6]	Yu [6]	SR+ANFIS [6]	SR+SVR [6]	Elman [6]	Cheng et al. [6]	PSO-RLSE	ABCO-RLSE
Best $\varphi$	0.012	0.012	0.012	0.012	0.012	0.012	0.002	0.002
利潤(TAIEX)	0	0	0	0	0	-231.02	1038.1	<b>1227.3</b>
利潤(HSI)	-1471	-1368	-602.94	190.71	<b>2342</b>	1793.12	355.04	503.59





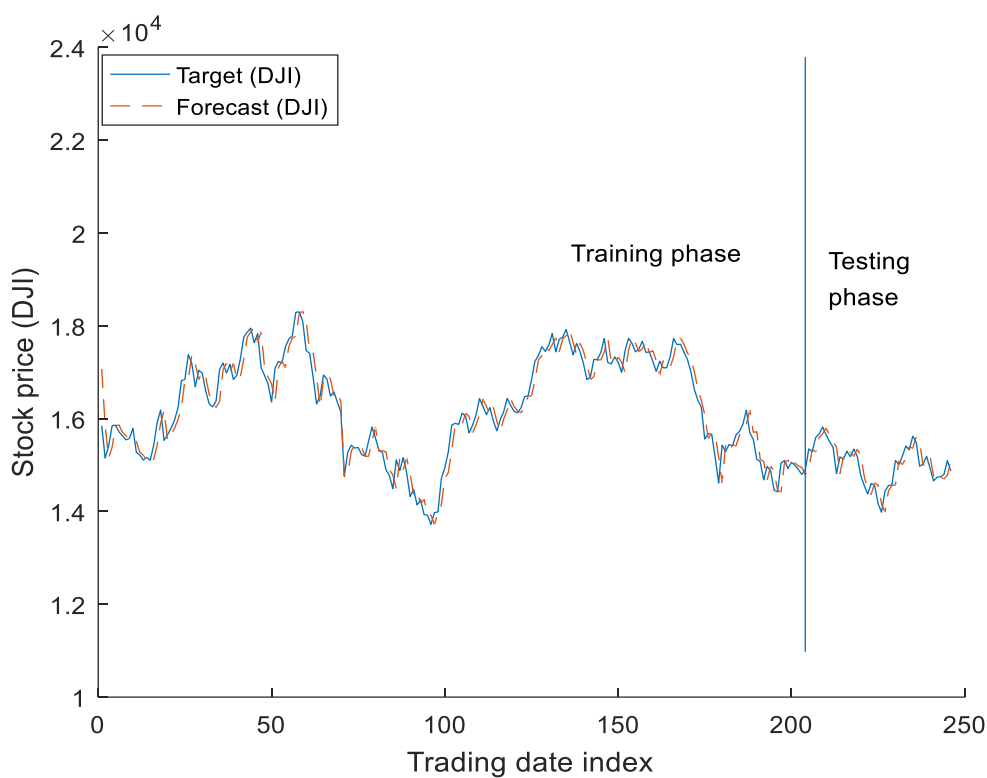
(a) TAIEX 預測(PSO-RLSE)



(b) TAIEX 預測(ABCO-RLSE)



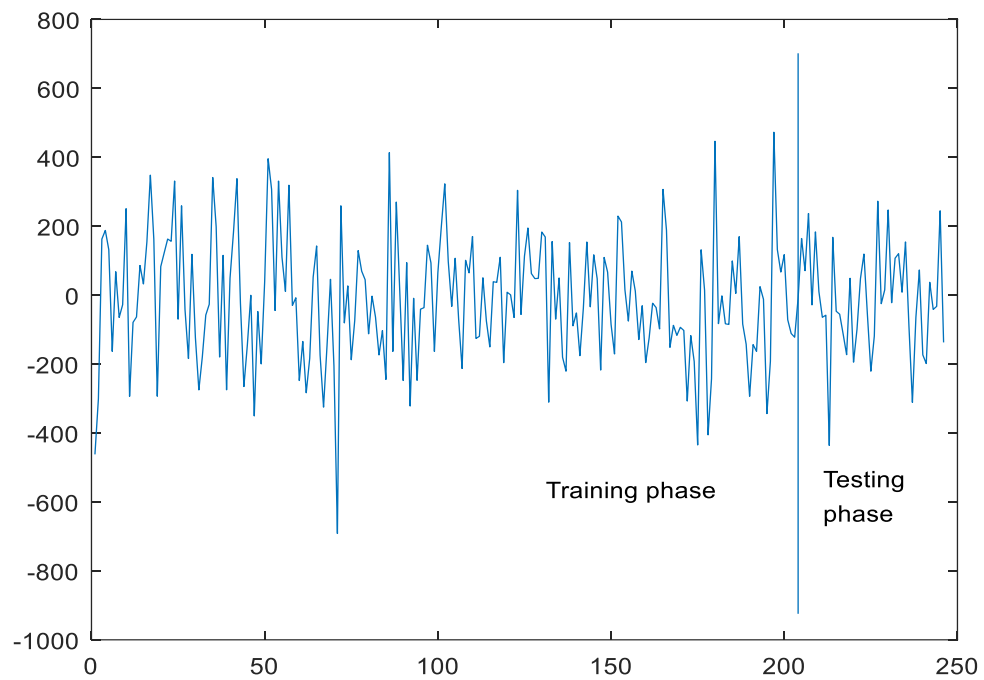
(c) DJI 預測(PSO-RLSE)



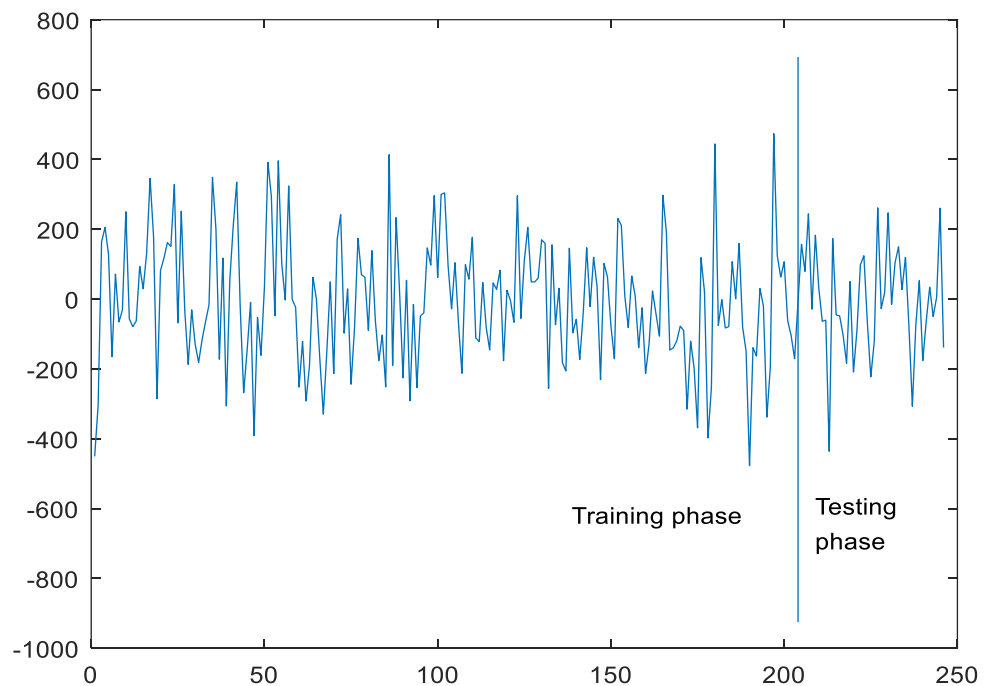
(d) DJI 預測(ABCO-RLSE)

圖 8 預測結果

藍色實線為實際目標值，紅色虛線為模型預測值，可以看出兩種演算法都有著不錯的預測效果



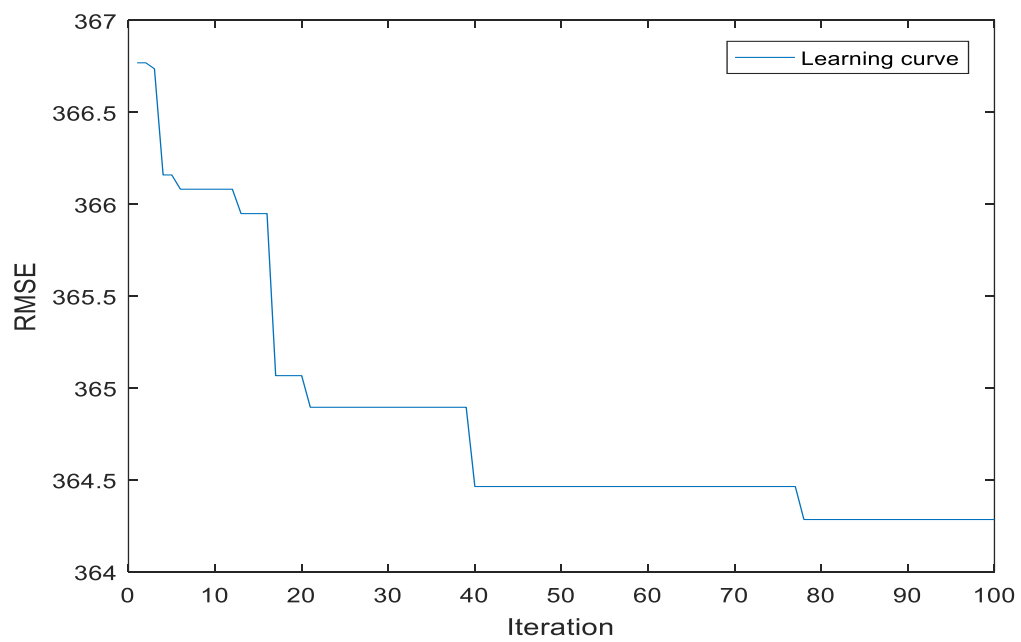
(a) PSO-RLSE



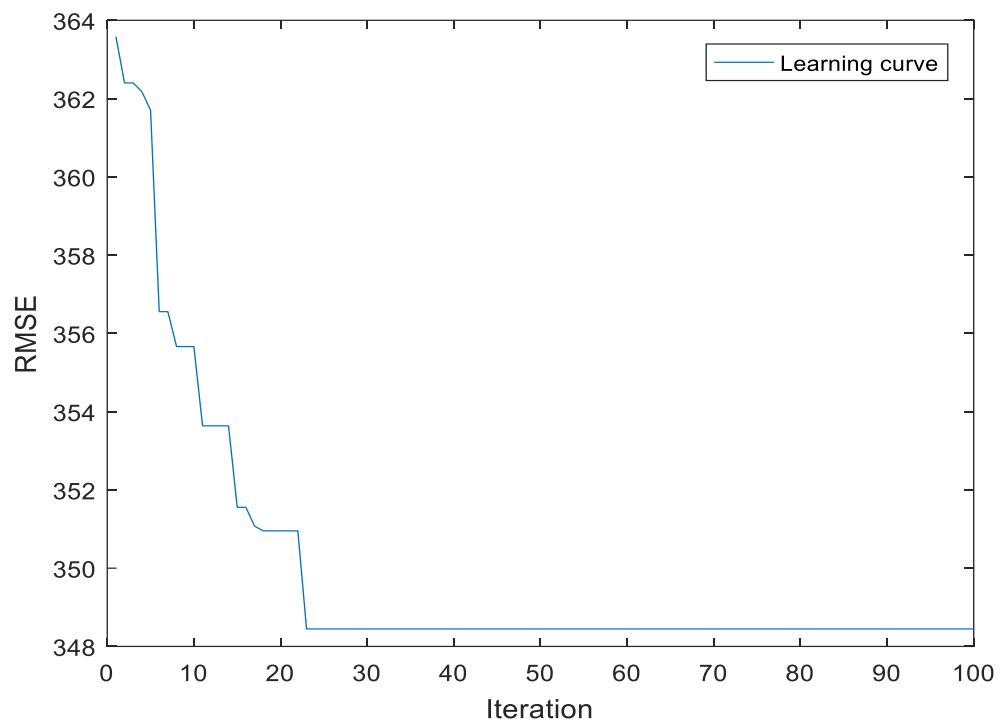
(b) ABCO-RLSE

圖 9 模型預測誤差值

兩種演算法的模型誤差呈現隨機亂數狀態，代表模型預測能力穩定



(a) PSO-RLSE



(b) ABCO-RLSE

圖 10 實驗二模型學習曲線

PSO-RLSE 在 75 回合時，模型逐漸穩定，ABCO-RLSE 在 25 回合左右模型逐漸穩定。

### 4.3 實驗三：台灣股票加權指數、道瓊工業指數、納斯達克和標準普爾 500 四目標預測

本實驗一次預測兩個目標，分別為 TAIEX 與 HSI。實驗目標是建立一適當的模型，經過訓練後預測每日股價指數。模型輸出為一複數值，取其實數部分預測 TAIEX，虛數部分預測 HSI。使用的資料為 TAIEX 2000 年的每日收盤價以及 HIS 2000 年的每日收盤價，此實驗的原始資料為 278 筆，經過一次差分後得到 277 筆，從每組資料中取出 30 個特徵值，共 60 個特徵，每個特徵為 247 筆資料，其中前 181 筆為訓練資料，剩餘為訓練資料，第 1 至 30 個特徵為 TAIEX 收盤價，第 31 至 60 為 DJI 收盤價，第 61 至 90 為 NASDAQ，第 91 至 120 為 S&P500。透過多目標特徵挑選後選出特徵  $\{f_{89}, f_3, f_{79}, f_{82}\}$  做為模型輸入資料。結構學習部分，會透過減數分群演算法分群，在進行區塊挑選如第三章第二小節所介紹，本實驗各個輸入維度分別  $\{3, 3, 3, 3\}$  個複數模糊集，進行區塊挑選後，從原本的 81 個  $\pi$  神經元篩選到剩下 9 個  $\pi$  神經元，藉由資料驅動(Data driven)概念適當地建構模型，每個複數模糊集有 3 個參數，其中包含中心、標準差以及相位頻率參數，總共有 12 個複數模糊集，故前鑑部參數數量為 36，後鑑部型態為 T-S function，因此總共會有  $K * (M + 1)$  個參數， $K$  為後鑑部數目， $M$  為模型輸入維度數目。整體模型在結構學習後的參數，如表 13 所示。機器學習參數設定，如表 14 所示。本次範例的結果除了兩個演算法比較也與其他論文 [27] 所提的方法做比較，像是 SVR、ANFIS、RBF 和 CNFS-ARIMA 等。比較結果如表 15 所示。為了測試模型的穩定性，我們總共跑了十次的實驗，效能統計結果如表 16 所示，訓練後的模型較容易表現出更佳的效能。模擬投資所賺的利潤，以及買賣次數如表 17 所示，標準差越高，代表投資的風險較高。目標與模型輸出的結果，如圖 11 所示；模型的機器學習曲線，如圖 12 所示；預測誤差如圖 13 所示。

表 13 實驗三模型設定

參數	值
特徵變數(輸入變數)	$\{f_{89}, f_3, f_{79}, f_{82}\}$
模糊集數量	$\{2, 3, 3, 3\}$
模糊集型態	CFS
複數型態目標數目	2
$\Pi$ 神經元數目	9
前鑑部參數數目	36
T-S 神經元數目	9
後鑑部參數數目	45

\* 每一個複數目標的實部與虛部，分別包含兩個實數目標

表 14 機器學習參數設定

<b>PSO</b>	
粒子群大小	50
迭代次數	100
$\{\omega, c_1, c_2\}$	$\{0.8, 2.0, 2.0\}$
$\xi_1, \xi_2$	Random in $[0,1]$
粒子初始化位置	藉由第三章所提 SC 演算法決定
粒子初始化速度	0
<b>ABCO</b>	
工蜂數量	40
觀察蜂數量	10
迭代次數	100
限制回合數	20
<b>RLSE</b>	
$\alpha$	$1 \times 10^8$
$\vec{\theta}_0$	25-by-1 全零向量
$\mathbf{P}_0$	$\alpha \mathbf{I}$
$\mathbf{I}$	25-by-25 單位矩陣

表 15 效能比較(實驗三)

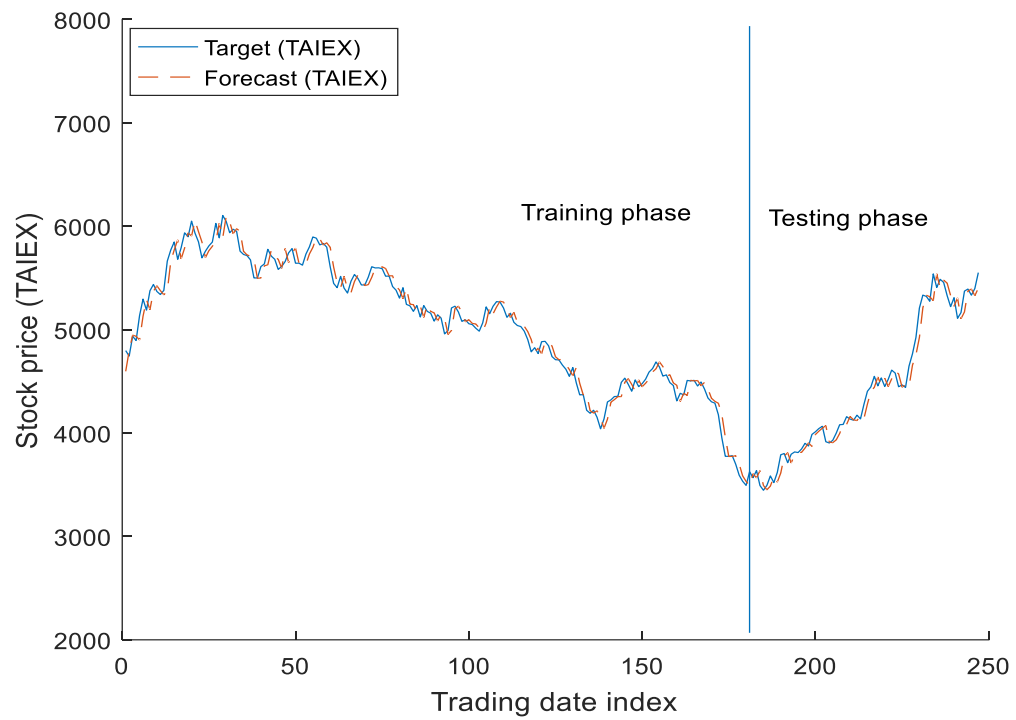
方法	TAIEX	DJI
SVR (two models, each with single output) [27]	162.46	101.44
ANFIS (two models, each with single output) [27]	147.36	105.56
ANFIS (one model with two outputs) [27]	151.62	128.20
RBF (two models, each with single output) [27]	134.32	106.33
RBF (one model with two outputs) [27]	137.58	181.79
CNFS(5)-ARIMA (one model with two outputs) [27]	115.82	103.06
PSO-RLSE	<b>101.61</b>	101.82
ABCO-RLSE	102.61	<b>100.79</b>

表 16 十次重複實驗效能統計(實驗三)

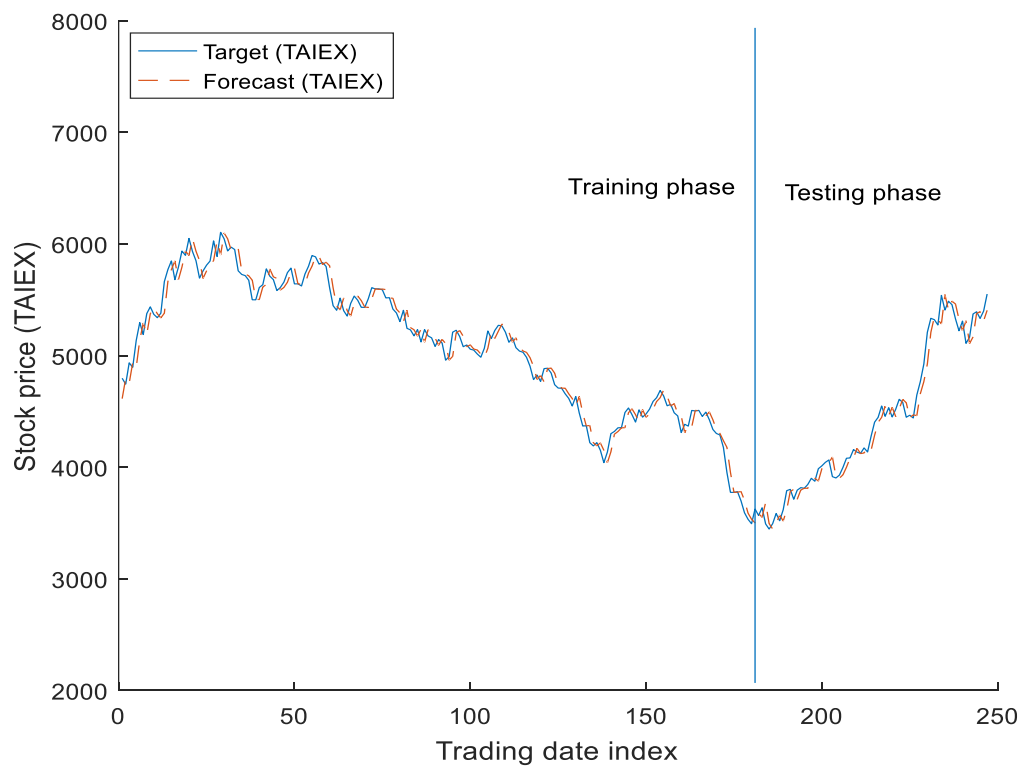
Trials	PSO-RLSE	ABCO-RLSE	PSO-RLSE*	ABCO-RLSE*
1	185.99	199.32	182.94	223.16
2	183.75	183.89	183.65	<b>183.14</b>
3	183.72	204.56	<b>182.69</b>	187.91
4	183.81	256.10	208.83	188.52
5	184.39	<b>180.82</b>	183.67	190.22
6	191.31	199.00	183.13	182.46
7	<b>181.90</b>	217.09	183.79	1178.7
8	281.103	182.13	183.63	188.60
9	183.68	188.24	185.35	183.53
10	183.67	183.89	183.20	183.69

表 17 模擬投資利潤表(實驗三)

Trials	PSO-RLSE		ABCO-RLSE		PSO-RLSE*		ABCO-RLSE*	
	利潤	買/賣	利潤	買/賣	利潤	買/賣	利潤	買/賣
Best $\varphi$	0.052		0.049		0.005		0.049	
1	1262.3	123/106	184.02	119/109	84.23	16/19	-163.3	123/103
2	175.28	122/107	587.86	122/106	355.64	15/21	588.84	121/96
3	403.52	122/108	603.22	124/105	292.45	16/21	2018.2	104/110
4	175.28	122/107	15.25	123/103	209.10	16/20	-473.1	120/103
5	-142.2	120/109	97.19	121/107	355.64	15/21	-87.73	119/109
6	543.69	123/108	228.15	122/107	292.45	16/21	21.14	120/105
7	-166.2	123/106	-230.5	121/106	355.64	15/21	152.62	126/102
8	-377.6	124/103	16.12	123/105	355.64	15/21	775.44	124/105
9	449.29	122/108	287.13	123/106	434.03	22/14	1026.1	122/106
10	446.29	122/108	587.76	122/106	454.22	16/21	191.28	124/104
平均	276.67		237.61		318.90		404.95	
標準差	463.88		283.16		108.26		726.69	

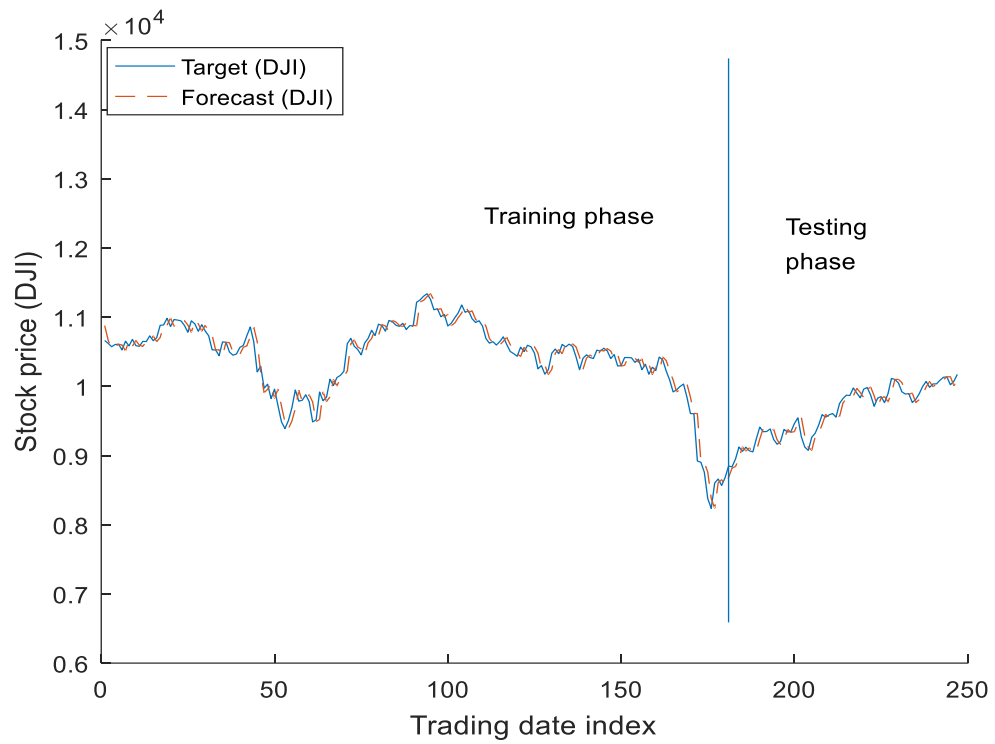


(a) TAIEX 預測(PSO-RLSE)

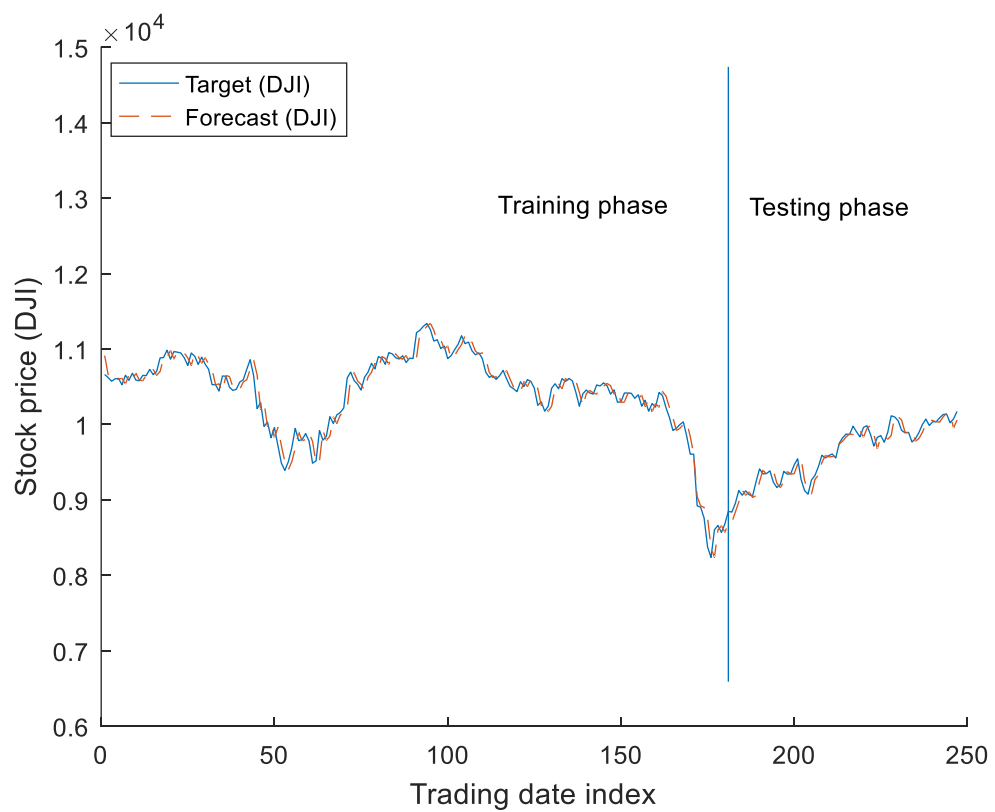


(b) TAIEX 預測(ABCO-RLSE)

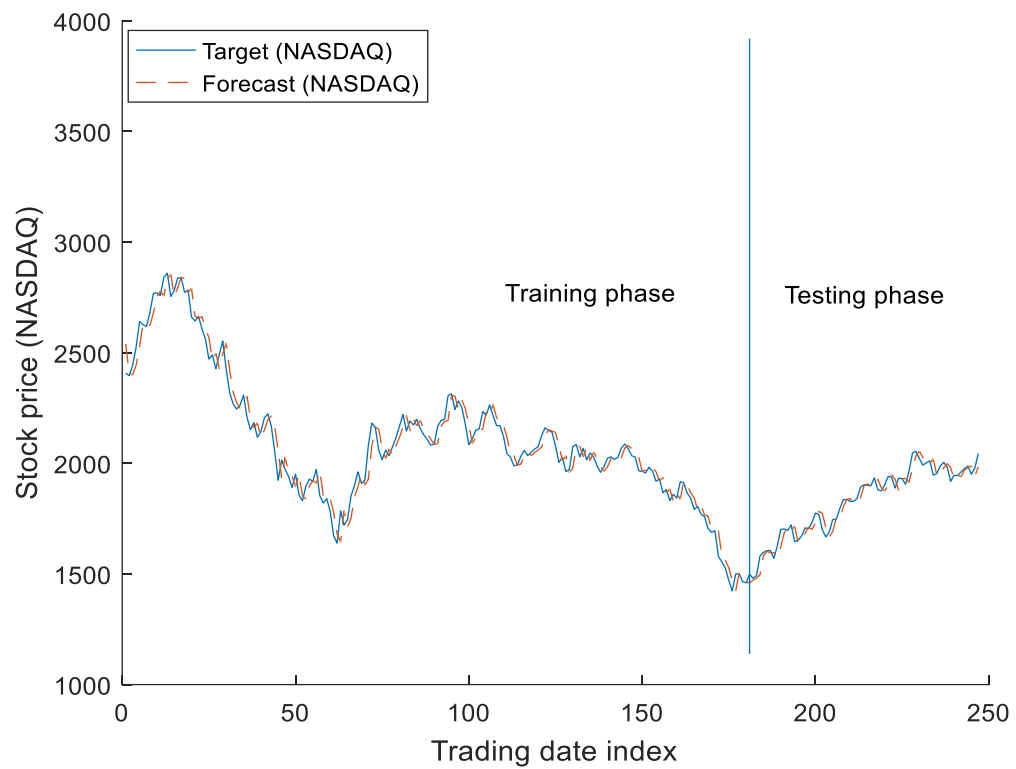




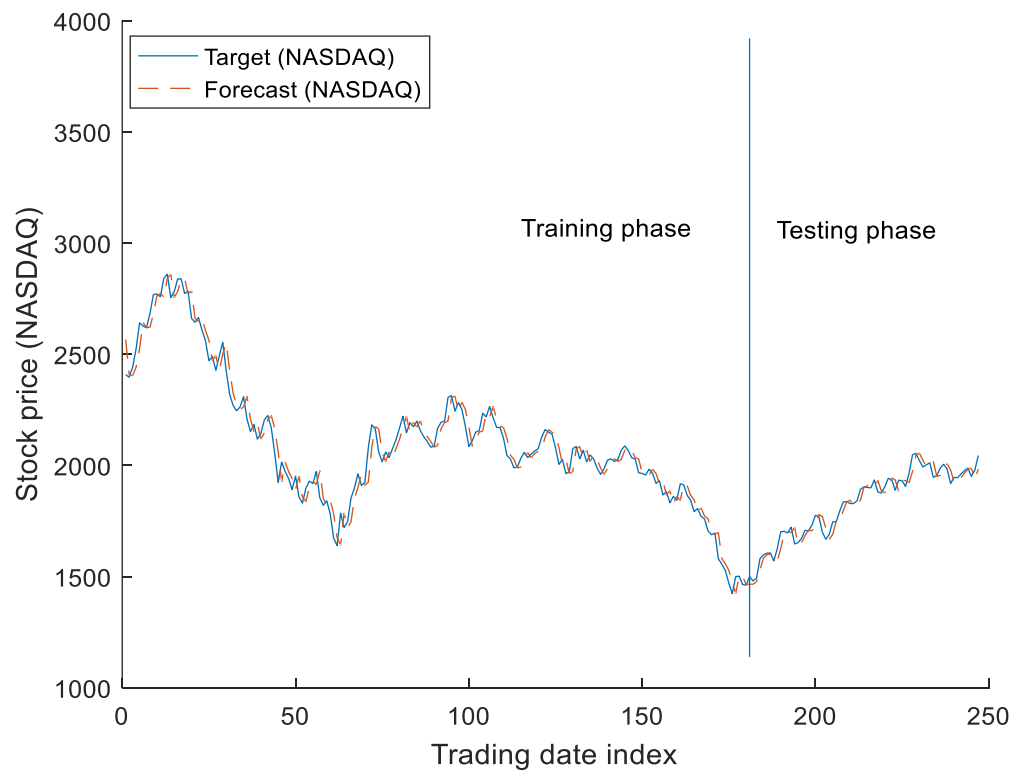
(c) DJI 預測(PSO-RLSE)



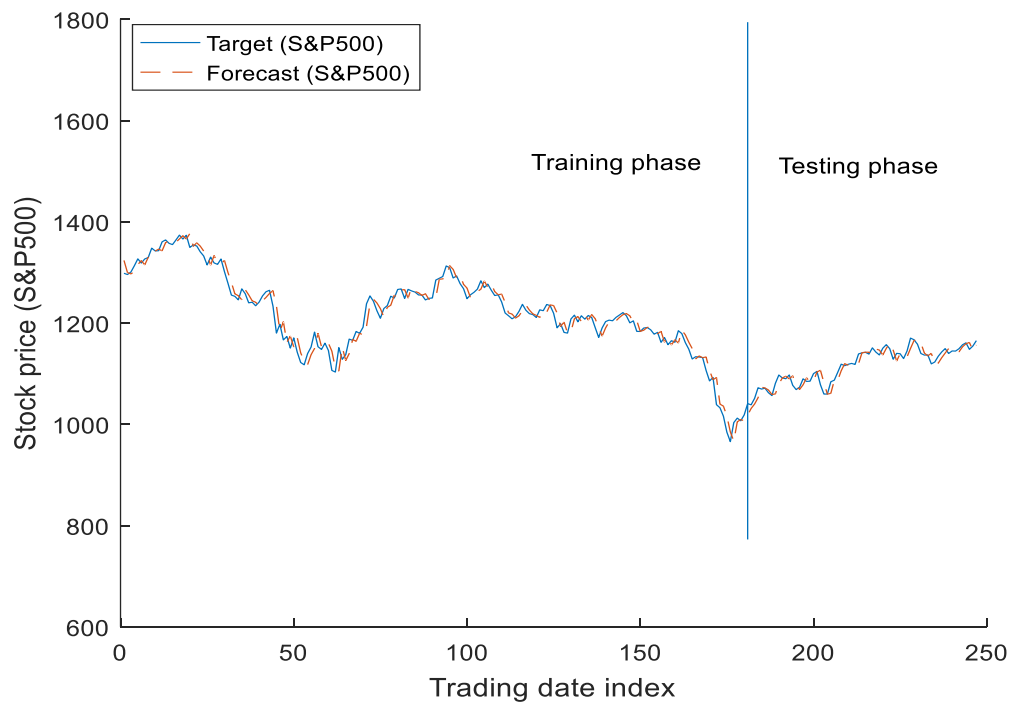
(d) DJI 預測(ABCO-RLSE)



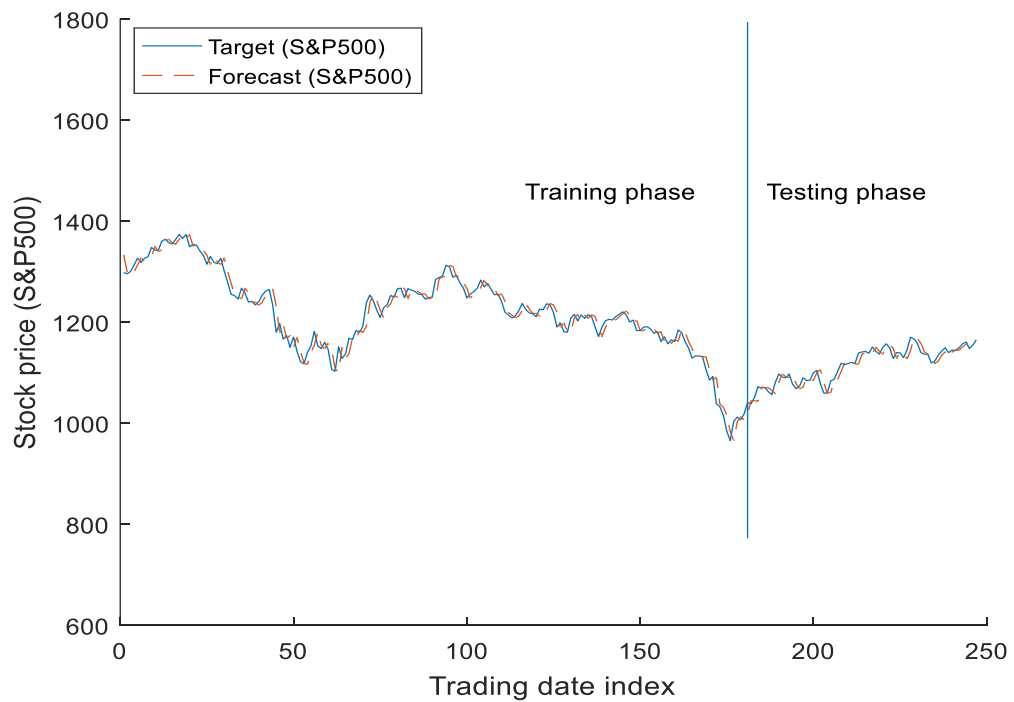
(e) NASDAQ 預測(PSO-RLSE)



(f) NASDAQ 預測(ABCO-RLSE)



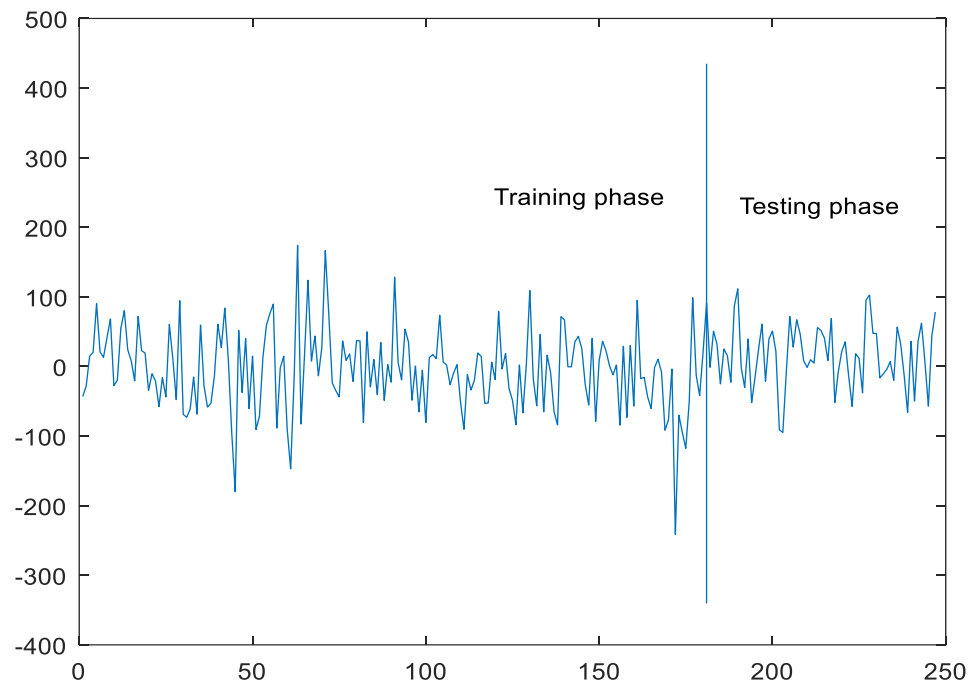
(g) S&amp;P500 預測(PSO-RLSE)



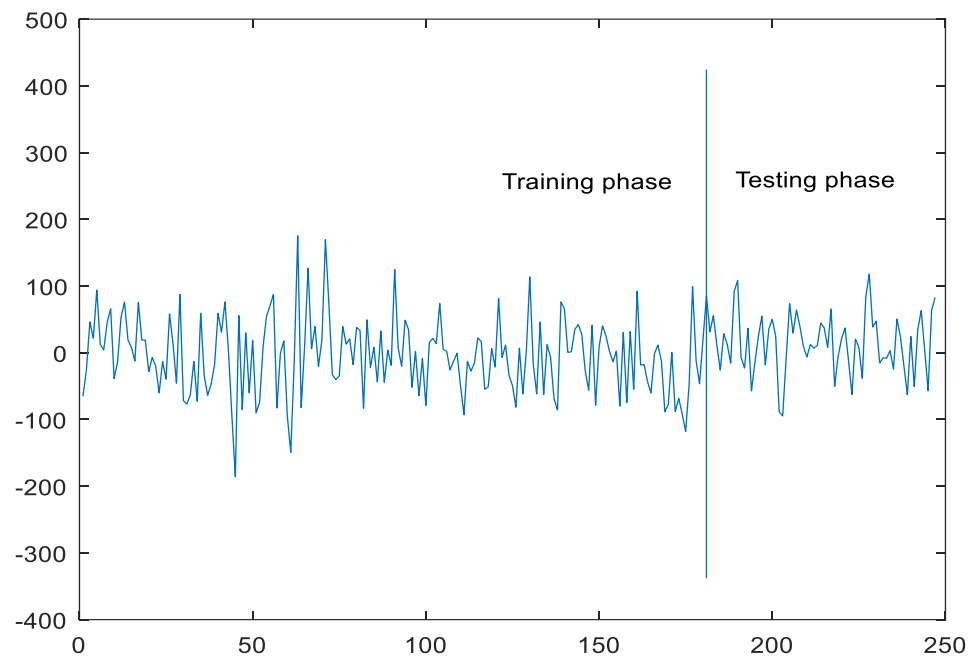
(h) S&amp;P500 預測(ABCO-RLSE)

圖 11 預測結果

藍色實線為實際目標值，紅色虛線為模型預測值，可以看出兩種演算法不論在訓練或是測試階段都有著不錯的預測效果



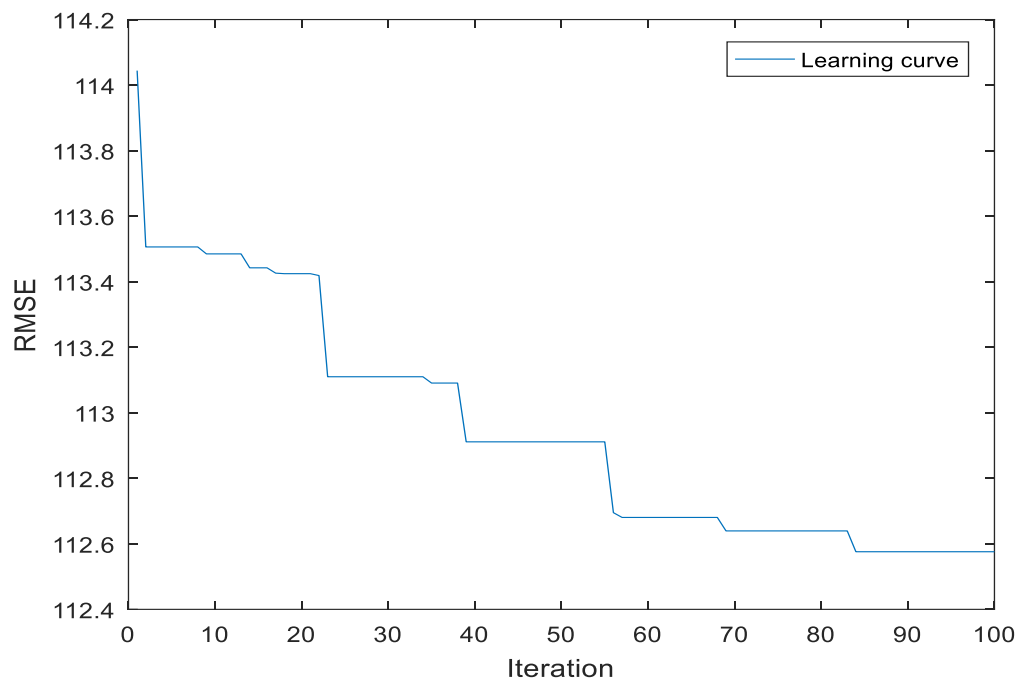
(a) PSO-RLSE



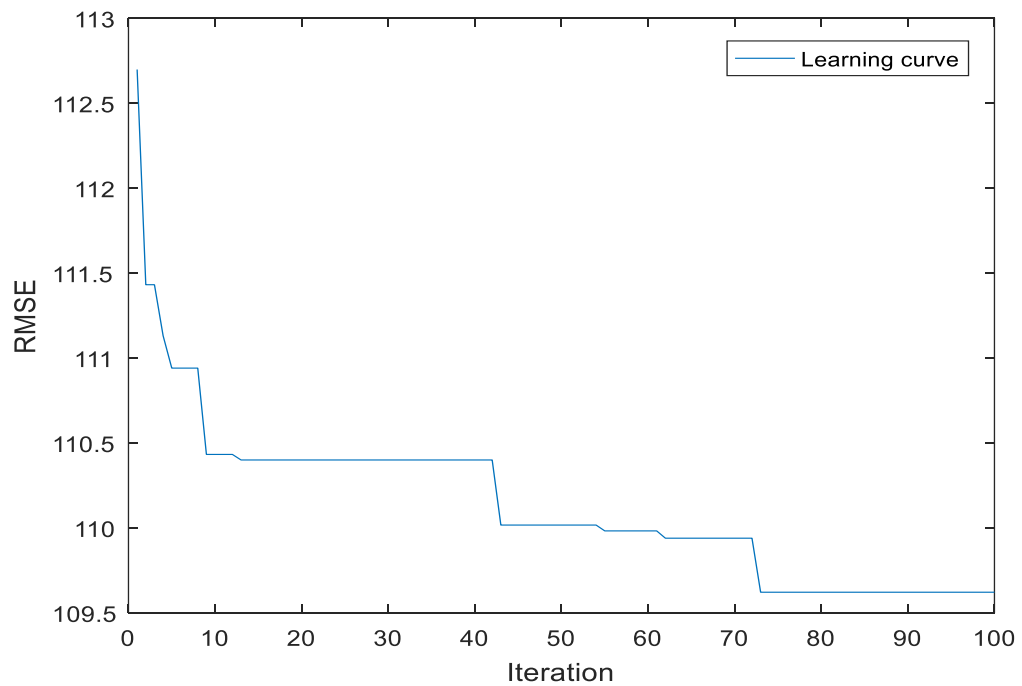
(b) ABCO-RLSE

圖 12 模型預測誤差值

預測誤差呈現隨機亂數狀態，介於-100~100 之間，代表模型預測能力穩定。



(a) PSO-RLSE



(b) ABCO-RLSE

圖 13 實驗三模型學習曲線

可以看出 PSO-RLSE 在 85 迭代時，模型逐漸穩定，ABCO-RLSE 在 10 回合時學習有些許的停滯，但之後又找到更好的位置直到 75 回合左右逐漸穩定。

## 第五章 討論

本研究提出以複合型演算法 PSO-RLSE 以及 ABCO-RLSE 優化模型參數，採用分治法降低演算法搜尋負擔，模型在建構時，藉由資料驅動的概念透過機器學習控制模型大小，並且在訓練資料進入模型前，事先進行特徵選取，選出對預測目標較為有利之特徵資料進行時間序列之預測。根據 3 個實驗的測試，驗證本研究提出的一系列方法對於時間序列的預測擁有較佳的預測性能。

本研究資料前處理採用多目標特徵挑選 [28]，取自夏農資訊熵 [35] 的理論。利用資料彼此之間的資訊熵以及資訊熵差量，計算資料候選特徵對於目標資料所能提供的資訊量，並考慮與已被選取特徵所造成的冗餘資訊量。最後從大量的資料中，篩選出對單目標或是多目標皆有用的特徵資料，以減少過多資料造成的負擔以及運算複雜度。

結構學習會決定模型的大小，首先透過減數分群演算法初步定位複數模糊集的初始位置，最後導入資料密度量概念，從大量的區塊中挑選出較少量的區塊作為規則數，降低模型的預算負擔，從 3 個實驗當中，可以發現原本的區塊數目個別從 144、81、81 降低至 15、9、9，顯然地減少模型運算時的負擔。規則數的建立也會影響到後鑑部層的數目，因此有效的縮減規則數，也會影響到參數學習時所需學習的參數個數，使其在參數搜尋過程中更容易找到最佳解。

複數類神經模糊系統以複數模糊集、T-S 模糊系統等方式建立。透過非線性的高斯複數模糊集以及線性的 T-S 模糊系統結合而成一非線性的模糊系統，以 If-Then 規則的型態產生神經元一對一的關係，使其系統可以較為人類所理解。複數模糊集相較於傳統的模糊集合，其歸屬程度從一維實數空間延伸到一個二維單位圓盤的平面空間，此方式使歸屬程度能夠容納更多的資訊，有助於提升模糊系統的推理能力與應用效能。而利用複數歸屬程度可以使模型系統產生 1 組複數型態的輸出，使其同時預測 2 個實數型態目標。而透過歸屬程度的拆解，可以得出總

共 3 個複數型態的歸屬程度，讓使原本的雙輸出預測能夠預測至 6 個目標，增強模型的預測能力。

複數類神經模糊模型的參數學習，前鑑部利用 PSO 與 ABCO 進行學習，兩者皆有使用到群體智慧的概念，透過群體內的資訊交換，以尋找更佳的位置，PSO 具有三個特性，第一，自動調整步伐，會透過權重與群體智慧調整位置；第二，隨機性，更新速度時有隨機變數的成分，有助於粒子活動性；第三，篤定性，每次在更新位置及速度時，會參照著群體最佳解以及自身最佳解方向移動；ABCO 則也具有三種特性，第一，隨機性，觀察蜂在選擇食物來源時，會使用到輪盤法，因此被選中的食物來源除了本身位置好，也帶有一定的隨機性；第二，局部搜尋，透過觀察蜂可以在食物源附近進型小部分搜尋避免錯過最佳解，第三，跳脫機制，透過是否有更新進行判斷食物源效用，若食物源一定回合內沒更新，則會派出偵查蜂取代該食物源位置，降低被困在局部最佳解 (Local minimum) 的機率。後鑑部是採用 RLSE 演算法，利用輸入資料點，以及前次的計算結果，尋找一線性函數，使資料點與該函數的平方誤差達到最小值，藉此找到符合資料關係的函數。並且利用不斷遞迴計算的方式，最佳化其後鑑部參數。

透過一系列的方法，可以看出本研究的 3 個實驗結果表現都優於其他方法，表示模型在單目標、雙目標以及多目標中，都有良好的預測能力，且有訓練的模型相較於沒訓練的模型，更容易有優秀的表現。在 3 個實驗中，可以發現 ABCO-RLSE 混合演算法的表現都比 PSO-RLSE 表現較好，代表局部搜尋以及跳脫機制的特性有著一定的效果，使得模型輸出更精確。而在投資策略下的虛擬投資中，我們有一個有趣的發現，模型的預測能力與虛擬投資利潤沒有正面相關，當一個模型的表現良好，並不代表虛擬投資效果也會良好，意即可以在十次的模型測試中，效能表現最好的測試與模擬投資利潤最高的並不相同，因為模型的曲線擬合代表著配市率，卻不代表能有效地賺取利潤，決定買與賣的操作才是關鍵因素，

舉例來說，一預測認為隔天會漲 100 元另一預測認為明天會跌 1 元，前者的策略

為買，後者的策略為賣，當明天實際價格為漲 1 元時，真正賺取利潤的為前者，但模型的預測誤差後者表現較優良。買賣的操作是透過投資策略中的門檻參數 $\varphi$ 所決定，此參數會因為模型與資料的不同影響到利潤，因此可在利潤比較表 6 與表 12 中發現本研究中 PSO-RLSE 與 ABCO-RLSE 的最佳 $\varphi$ 不盡相同，當然也與其他文獻不一樣。股市的漲跌預測仍然是具有一定難度，但從標準差可以看出模型對於投資利潤的穩定度，從表 5、表 11 以及表 17 中可以發現，有訓練過的模型十次模擬投資中的標準差較低，其中 ABCO-RLSE 又比 PSO-RLSE 低，意即預測能力較佳的模型，在投資的穩定性會比較高，風險相對來說較低。



## 第六章 結論與未來研究方向

### 6.1 結論

本研究提出一新型態混合演算法 ABCO-RLSE，結合了人工蜂群演算法以及遞迴最小平方法，用於優化複數類神經模糊模型系統的參數集合。而模型結合複數型模糊集合、T-S 模型系統以及類神經網路。於資料進入模型前輔以特徵資料選取，避免冗餘的輸入資料進入模型，耗費運算效能，減少過多資料對模型的負擔。採用複數模糊集使模型能夠有多個複數型態輸出，因為透過複數歸屬程度的解構，可將歸屬程度的實數部與虛數部取出各自成為另一歸屬程度，此舉使模型可產生三個複數輸出，最多預測六個實數目標，使得模型有別於傳統模糊系統能夠有同時進行多目標預測的能力。實驗二與實驗三證明此方式的貢獻。在結構學習方面，模型確實有效的縮減大小，並透過資料密度量的概念，選取對資料較有用的區塊，客觀地使用資料建造模型，避免複雜的運算，提升模型效率。參數學習時，ABCO 因為跳脫機制以及局部搜尋特性，可以比 PSO 更容易找到最佳解，表現效果優於其他文獻所提出的方法。投資策略透過門檻值的配合有著很大的賺錢機率，且發現訓練後以及效能較好的模型，十次模擬投資利潤的標準差較低，有助於降低投資者的風險，提供給投資者客觀的參考資料。

### 6.2 未來研究方向

透過本研究的實驗可以發現模型效能表現比其他文獻方法更加優秀，證明 PSO 與 ABCO 的最佳化能力，但此兩種演算法仍然有著自身的限制，如 PSO 有著早熟(premature)的缺點，意即在迭代次數的前幾回合模型就收斂，導致模型容易掉入區域最佳解；ABCO 在搜尋時則是限制於局部最佳解，觀察蜂會在食物源附近搜尋，當食物源更新效果不好，則整體效果也會不好，同時所需要的計算時間也比 PSO 演算法更長。因此未來可以使用其他演算法，配合複數神經模糊系統，

嘗試尋找更良好的效能表現。在投資策略方面，對於不同模型，我們發現最好的門檻參數會有所不同，因此未來可以透過機器學習演算法，最佳化門檻參數，或根據預測的漲跌調整投資時的權重，以達到更精確的投資操作，從股市中賺取利潤。

## 參考文獻

- [1] E. Abbasi and A. Abouec, "Stock price forecast by using neuro-fuzzy inference system," in *Proceedings of World Academy of Science, Engineering and Technology*, 2008, vol. 36, pp. 320-323: Citeseer.
- [2] P. L. Avrim L. Blum, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1-2, 1997.
- [3] J. Buckley, "Fuzzy complex numbers," *Fuzzy Sets and Systems*, vol. 33, no. 3, pp. 333-345, 1989.
- [4] J. Buckley and Y. Qu, "Fuzzy complex analysis I: differentiation," *Fuzzy Sets and Systems*, vol. 41, no. 3, pp. 269-284, 1991.
- [5] J. J. Buckley, "Fuzzy complex analysis II: integration," *Fuzzy Sets and Systems*, vol. 49, no. 2, pp. 171-179, 1992.
- [6] C.-H. Cheng and J.-H. Yang, "Fuzzy time-series model based on rough set rule induction for forecasting stock price," *Neurocomputing*, vol. 302, pp. 33-45, 2018.
- [7] S. L. Chiu, "Fuzzy model identification based on cluster estimation," *Journal of Intelligent & fuzzy systems*, vol. 2, no. 3, pp. 267-278, 1994.
- [8] R. Clausius, "Über eine veränderte Form des zweiten Hauptsatzes der mechanischen Wärmetheorie," *Annalen der Physik*, vol. 169, no. 12, pp. 481-506, 1854.
- [9] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273-297, 1995.
- [10] L. Davis, "Handbook of genetic algorithms," 1991.
- [11] E. F. Fama, "Random walks in stock market prices," *Financial analysts journal*, vol. 51, no. 1, pp. 75-80, 1995.
- [12] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157-1182, 2003.
- [13] D. O. Hebb, "The organization of behavior: A neuropsychological theory," ed: New York: Wiley, 1949.
- [14] N. Hoque, D. Bhattacharyya, and J. K. Kalita, "MIFS-ND: a mutual information-based feature selection method," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6371-6385, 2014.
- [15] H.-H. Hsu, C.-W. Hsieh, and M.-D. Lu, "Hybrid feature selection by combining filters and wrappers," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8144-8150, 2011.
- [16] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing; a computational approach to learning and machine intelligence," 1997.

- [17] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer engineering department 2005.
- [18] M. G. Kendall and A. B. Hill, "The analysis of economic time-series-part i: Prices," *Journal of the Royal Statistical Society. Series A (General)*, vol. 116, no. 1, pp. 11-34, 1953.
- [19] J. Kennedy and R. Eberhart, "PSO optimization," in *Proc. IEEE Int. Conf. Neural Networks*, 1995, vol. 4, pp. 1941-1948: IEEE Service Center, Piscataway, NJ.
- [20] K.-j. Kim and I. Han, "Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index," *Expert systems with Applications*, vol. 19, no. 2, pp. 125-132, 2000.
- [21] T. Kimoto, K. Asakawa, M. Yoda, and M. Takeoka, "Stock market prediction system with modular neural networks," in *Neural Networks, 1990., 1990 IJCNN International Joint Conference on*, 1990, pp. 1-6: IEEE.
- [22] R. S. Koijen, H. Lustig, and S. Van Nieuwerburgh, "The cross-section and time series of stock and bond returns," *Journal of Monetary Economics*, vol. 88, pp. 50-69, 2017.
- [23] C. Li and T.-W. Chiang, "Complex neuro-fuzzy self-learning approach to function approximation," in *Asian Conference on Intelligent Information and Database Systems*, 2010, pp. 289-299: Springer.
- [24] C. Li and T.-W. Chiang, "Complex fuzzy computing to time series prediction—A multi-swarm PSO learning approach," in *Asian Conference on Intelligent Information and Database Systems*, 2011, pp. 242-251: Springer.
- [25] C. Li, C. W. Lin, and H. Huang, "Neural Fuzzy Forecasting of the China Yuan to US Dollar Exchange Rate—A Swarm Intelligence Approach," in *International Conference in Swarm Intelligence*, 2011, pp. 616-625: Springer.
- [26] C. Li and J.-W. Hu, "A new ARIMA-based neuro-fuzzy approach and swarm intelligence for time series forecasting," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 2, pp. 295-308, 2012.
- [27] C. Li and T.-W. Chiang, "Complex neurofuzzy ARIMA forecasting—a new approach using complex fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 3, pp. 567-584, 2013.
- [28] C. Li, "國立中央大學資訊管理所李俊賢教授, 研究生訓練課程內容," ed, 2016-2018.
- [29] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115-133, 1943.

- [30] Z. Pan and L. Liu, "Forecasting stock return volatility: A comparison between the roles of short-term and long-term leverage effects," *Physica A: Statistical Mechanics and its Applications*, vol. 492, pp. 168-180, 2018.
- [31] P. B. Patel and T. Marwala, "Neural networks, fuzzy inference systems and adaptive-neuro fuzzy inference systems for financial decision making," in *International Conference on Neural Information Processing*, 2006, pp. 430-439: Springer.
- [32] D. Ramot, R. Milo, M. Friedman, and A. Kandel, "Complex fuzzy sets," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 171-186, 2002.
- [33] N. Rochester, J. Holland, L. Haibt, and W. Duda, "Tests on a cell assembly theory of the action of the brain, using a large digital computer," *IRE Transactions on information Theory*, vol. 2, no. 3, pp. 80-93, 1956.
- [34] N. Russel, "Artificial Intelligenz-A Modern Approach, 2nd," ed: Ed.
- [35] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3-55, 2001.
- [36] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," in *Readings in Fuzzy Sets for Intelligent Systems*: Elsevier, 1993, pp. 387-403.
- [37] J. Vieira, F. M. Dias, and A. Mota, "Neuro-fuzzy systems: a survey," in *5th WSEAS NNA international conference on neural networks and applications, Udine, Italia*, 2004.
- [38] L.-Y. Wei, "A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting," *Applied Soft Computing*, vol. 42, pp. 368-376, 2016.
- [39] P. Werbos, "Beyond regression: new fools for prediction and analysis in the behavioral sciences," *PhD thesis, Harvard University*, 1974.
- [40] J. Yao, C. L. Tan, and H.-L. Poh, "Neural networks for technical analysis: a study on KLCI," *International journal of theoretical and applied finance*, vol. 2, no. 02, pp. 221-241, 1999.
- [41] S. Yao, M. Pasquier, and C. Quek, "A foreign exchange portfolio management mechanism based on fuzzy neural networks," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, 2007, pp. 2576-2583: IEEE.
- [42] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *Journal of machine learning research*, vol. 5, no. Oct, pp. 1205-1224, 2004.
- [43] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338-353, 1965.
- [44] X. Zhi-Bin and L. Rong-Jun, "Credit risk evaluation with fuzzy neural networks

on listed corporations of China," in *VLSI Design and Video Technology, 2005. Proceedings of 2005 IEEE International Workshop on*, 2005, pp. 397-402: IEEE.